

Εγχειρίδιο Χρήσης Family Doctor

1. Εισαγωγή

Η εφαρμογή Family Doctor επιτρέπει σε πολίτες να δηλώνουν τα στοιχεία της οικογένειάς τους και να υποβάλλουν αιτήσεις στον οικογενειακό τους γιατρό. Ο οικογενειακός γιατρός μπορεί στη συνέχεια να αποδεχτεί ή να απορρίψει τις αιτήσεις.

2. Βασικές Λειτουργίες Πολίτη

2.1 Προβολή Λίστας Πολιτών

Η ενότητα `/citizen` παρέχει δυνατότητα προβολής λίστας με όλους τους πολίτες που έχουν δηλωθεί.

- **Διαδρομή:** `/citizen`
- **Μέθοδος:** `GET`
- **Αναμενόμενη Απόκριση:**
 - Εμφάνιση λίστας με τα στοιχεία των πολιτών.

2.2 Προσθήκη Νέου Πολίτη

Η ενότητα `/citizen/new` επιτρέπει την προσθήκη νέου πολίτη στο σύστημα.

- **Διαδρομή:** `/citizen/new`
- **Μέθοδος:** `GET`
- **Αναμενόμενη Απόκριση:**
 - Φόρμα για την εισαγωγή των στοιχείων του νέου πολίτη.

2.3 Επεξεργασία Πληροφοριών Πολίτη

Η ενότητα `/citizen/{citizen_id}` επιτρέπει στους διαχειριστές την επεξεργασία των πληροφοριών ενός πολίτη.

- **Διαδρομή:** `/citizen/{citizen_id}`
- **Μέθοδος:** `GET`
- **Αναμενόμενη Απόκριση:**
 - Φόρμα επεξεργασίας με τα ήδη υπάρχοντα στοιχεία του πολίτη.

2.4 Αποθήκευση Νέου Πολίτη

Η ενότητα `/citizen/new` διαχειρίζεται την αποθήκευση των νέων πληροφοριών πολίτη.

- **Διαδρομή:** `/citizen/new`
- **Μέθοδος:** `POST`
- **Αναμενόμενη Απόκριση:**

- Ενημέρωση της λίστας με τους πολίτες μετά την προσθήκη του νέου πολίτη.

2.5 Διαγραφή Πολίτη

Η ενότητα `/citizen/{citizen_id}/delete` επιτρέπει τη διαγραφή ενός πολίτη.

- **Διαδρομή:** `/citizen/{citizen_id}/delete`
- **Μέθοδος:** `GET`
- **Αναμενόμενη Απόκριση:**

- Διαγραφή του πολίτη και των σχετικών αιτήσεων.

2.6 Προβολή Αιτημάτων Πολίτη

Η ενότητα `/citizen/{citizen_id}/request` επιτρέπει την προβολή όλων των αιτημάτων που έχει υποβάλει ένας πολίτης.

- **Διαδρομή:** `/citizen/{citizen_id}/request`
- **Μέθοδος:** `GET`
- **Αναμενόμενη Απόκριση:**

- Προβολή λίστας με τα αιτήματα του πολίτη.

2.7 Επεξεργασία Αιτήματος Πολίτη

Η ενότητα `/citizen/{citizen_id}/request/{request_id}` επιτρέπει στους διαχειριστές την επεξεργασία ενός συγκεκριμένου αιτήματος πολίτη.

- **Διαδρομή:** `/citizen/{citizen_id}/request/{request_id}`
- **Μέθοδος:** `GET`
- **Περιγραφή:**

- Εμφάνιση φόρμας επεξεργασίας με τα ήδη υπάρχοντα στοιχεία του αιτήματος πολίτη.

2.8 Αποθήκευση Ενημερωμένου Αιτήματος Πολίτη

Η ενότητα `/citizen/{citizen_id}/request/{request_id}` διαχειρίζεται την αποθήκευση των ενημερωμένων πληροφοριών αιτήματος πολίτη.

- **Διαδρομή:** `/citizen/{citizen_id}/request/{request_id}`
- **Μέθοδος:** POST
- **Περιγραφή:**
 - Ενημέρωση του αιτήματος πολίτη με βάση τα νέα δεδομένα.
 - Επιστροφή στη σελίδα πολιτών μετά την επεξεργασία.

2.9 Προσθήκη Νέου Αιτήματος Πολίτη

Η ενότητα `/citizen/{citizen_id}/request/new` επιτρέπει την προσθήκη νέου αιτήματος πολίτη.

- **Διαδρομή:** `/citizen/{citizen_id}/request/new`
- **Μέθοδος:** GET
- **Περιγραφή:**
 - Εμφάνιση φόρμας για την εισαγωγή νέου αιτήματος πολίτη.

2.10 Αποθήκευση Νέου Αιτήματος Πολίτη

Η ενότητα `/citizen/{citizen_id}/request/new` διαχειρίζεται την αποθήκευση του νέου αιτήματος πολίτη.

- **Διαδρομή:** `/citizen/{citizen_id}/request/new`
- **Μέθοδος:** POST
- **Περιγραφή:**
 - Αποθήκευση του νέου αιτήματος πολίτη στο σύστημα.
 - Επιστροφή στη σελίδα πολιτών μετά την προσθήκη.

2.11 Διαγραφή Αιτήματος Πολίτη

Η ενότητα `/citizen/{citizen_id}/request/{request_id}/delete` επιτρέπει τη διαγραφή ενός συγκεκριμένου αιτήματος πολίτη.

- **Διαδρομή:** `/citizen/{citizen_id}/request/{request_id}/delete`
- **Μέθοδος:** GET
- **Περιγραφή:**
 - Διαγραφή του συγκεκριμένου αιτήματος πολίτη από το σύστημα.
 - Επιστροφή στη σελίδα αιτημάτων πολίτη μετά τη διαγραφή.

3. Βασικές Λειτουργίες Ιατρού

3.1 Προβολή Λίστας Ιατρών

Η ενότητα `/doctor` παρέχει δυνατότητα προβολής λίστας με όλους τους ιατρούς που υπάρχουν στο σύστημα.

- **Διαδρομή:** `/doctor`
- **Μέθοδος:** GET
- **Αναμενόμενη Απόκριση:**
 - Εμφάνιση λίστας με τα στοιχεία των ιατρών.

3.2 Προσθήκη Νέου Ιατρού

Η ενότητα `/doctor/new` επιτρέπει την προσθήκη νέου ιατρού στο σύστημα.

- **Διαδρομή:** `/doctor/new`
- **Μέθοδος:** GET
- **Αναμενόμενη Απόκριση:**
 - Φόρμα για την εισαγωγή των στοιχείων του νέου ιατρού.

3.3 Επεξεργασία Πληροφοριών Ιατρού

Η ενότητα `/doctor/{doctor_id}` επιτρέπει στους διαχειριστές την επεξεργασία των πληροφοριών ενός συγκεκριμένου ιατρού.

- **Διαδρομή:** `/doctor/{doctor_id}`
- **Μέθοδος:** `GET`
- **Αναμενόμενη Απόκριση:**
 - Φόρμα επεξεργασίας με τα ήδη υπάρχοντα στοιχεία του ιατρού.

3.4 Αποθήκευση Νέου Ιατρού

Η ενότητα `/doctor/new` διαχειρίζεται την αποθήκευση των νέων πληροφοριών ιατρού.

- **Διαδρομή:** `/doctor/new`
- **Μέθοδος:** `POST`
- **Αναμενόμενη Απόκριση:**
 - Ενημέρωση της λίστας με τους ιατρούς μετά την προσθήκη του νέου ιατρού.

3.5 Διαγραφή Ιατρού

Η ενότητα `/doctor/{doctor_id}/delete` επιτρέπει τη διαγραφή ενός συγκεκριμένου ιατρού.

- **Διαδρομή:** `/doctor/{doctor_id}/delete`
- **Μέθοδος:** `GET`
- **Αναμενόμενη Απόκριση:**
 - Διαγραφή του ιατρού και των σχετικών αιτήσεων.

3.6 Προβολή Αιτημάτων Ιατρού

Η ενότητα `/doctor/{doctor_id}/request` επιτρέπει την προβολή όλων των αιτημάτων που έχει λάβει ένας ιατρός.

- **Διαδρομή:** `/doctor/{doctor_id}/request`
- **Μέθοδος:** `GET`
- **Αναμενόμενη Απόκριση:**
 - Προβολή λίστας με τα αιτήματα του ιατρού.

3.7 Επεξεργασία Αιτήματος Ιατρού

Η ενότητα `/doctor/{doctor_id}/request/{request_id}` επιτρέπει στους ιατρούς την επεξεργασία ενός συγκεκριμένου αιτήματος.

- **Διαδρομή:** `/doctor/{doctor_id}/request/{request_id}`
- **Μέθοδος:** GET
- **Αναμενόμενη Απόκριση:**
 - Εμφάνιση φόρμας επεξεργασίας με τα ήδη υπάρχοντα στοιχεία του αιτήματος.

3.8 Αποθήκευση Ενημερωμένου Αιτήματος Ιατρού

Η ενότητα `/doctor/{doctor_id}/request/{request_id}` διαχειρίζεται την αποθήκευση των ενημερωμένων πληροφοριών αιτήματος.

- **Διαδρομή:** `/doctor/{doctor_id}/request/{request_id}`
- **Μέθοδος:** POST
- **Αναμενόμενη Απόκριση:**
 - Ενημέρωση του αιτήματος με βάση τα νέα δεδομένα.
 - Επιστροφή στη σελίδα αιτημάτων ιατρού μετά την επεξεργασία.

3.9 Διαγραφή Αιτήματος Ιατρού

Η ενότητα `/doctor/{doctor_id}/request/{request_id}/delete` επιτρέπει τη διαγραφή ενός συγκεκριμένου αιτήματος ιατρού.

- **Διαδρομή:** `/doctor/{doctor_id}/request/{request_id}/delete`
- **Μέθοδος:** GET
- **Αναμενόμενη Απόκριση:**
 - Διαγραφή του συγκεκριμένου αιτήματος ιατρού από το σύστημα.
 - Επιστροφή στη σελίδα αιτημάτων ιατρού μετά τη διαγραφή.

3.10 Αποδοχή Αιτήματος Ιατρού

Η ενότητα `/doctor/{doctor_id}/request/{request_id}/accept` επιτρέπει στους ιατρούς την αποδοχή ενός συγκεκριμένου αιτήματος.

- **Διαδρομή:** `/doctor/{doctor_id}/request/{request_id}/accept`
- **Μέθοδος:** `GET`
- **Αναμενόμενη Απόκριση:**

- Αποδοχή του συγκεκριμένου αιτήματος από τον ιατρό.
- Επιστροφή στη σελίδα αιτημάτων ιατρού.

3.11 Απόρριψη Αιτήματος Ιατρού

Η ενότητα `/doctor/{doctor_id}/request/{request_id}/decline` επιτρέπει στους ιατρούς την απόρριψη ενός συγκεκριμένου αιτήματος.

- **Διαδρομή:** `/doctor/{doctor_id}/request/{request_id}/decline`
- **Μέθοδος:** `GET`
- **Αναμενόμενη Απόκριση:**

- Απόρριψη του συγκεκριμένου αιτήματος από τον ιατρό.
- Επιστροφή στη σελίδα αιτημάτων ιατρού.

4. Χρήστες

4.1 Εγγραφή Νέου Χρήστη

Η ενότητα `/register` επιτρέπει τη δημιουργία νέου χρήστη στο σύστημα.

- **Διαδρομή:** `/register`
- **Μέθοδος:** `GET`
- **Αναμενόμενη Απόκριση:**

- Φόρμα για την εισαγωγή των πληροφοριών του νέου χρήστη.

4.2 Αποθήκευση Νέου Χρήστη

Η ενότητα `/saveUser` διαχειρίζεται την αποθήκευση των πληροφοριών ενός νέου χρήστη.

- **Διαδρομή:** `/saveUser`
- **Μέθοδος:** `POST`
- **Αναμενόμενη Απόκριση:**

- Αποθήκευση του νέου χρήστη στο σύστημα.
- Επιστροφή στην αρχική σελίδα.

4.3 Προβολή Λίστας Χρηστών

Η ενότητα `/users` παρέχει τη δυνατότητα προβολής λίστας με όλους τους χρήστες που υπάρχουν στο σύστημα.

- **Διαδρομή:** `/users`
- **Μέθοδος:** `GET`
- **Αναμενόμενη Απόκριση:**

- Προβολή λίστας με τα στοιχεία των χρηστών και των ρόλων τους.

4.4 Προβολή Στοιχείων Συγκεκριμένου Χρήστη

Η ενότητα `/user/{user_id}` επιτρέπει την προβολή των στοιχείων ενός συγκεκριμένου χρήστη.

- **Διαδρομή:** `/user/{user_id}`
- **Μέθοδος:** `GET`
- **Αναμενόμενη Απόκριση:**

- Προβολή στοιχείων του συγκεκριμένου χρήστη.

4.5 Αποθήκευση Ενημερωμένων Στοιχείων Χρήστη

Η ενότητα `/user/{user_id}` διαχειρίζεται την αποθήκευση των ενημερωμένων πληροφοριών ενός συγκεκριμένου χρήστη.

- **Διαδρομή:** `/user/{user_id}`
- **Μέθοδος:** POST
- **Αναμενόμενη Απόκριση:**
 - Ενημέρωση των πληροφοριών του χρήστη με βάση τα νέα δεδομένα.
 - Επιστροφή στην αρχική σελίδα λίστας χρηστών.

4.6 Προσθήκη Ρόλου σε Χρήστη

Η ενότητα `/user/role/add/{user_id}/{role_id}` επιτρέπει την προσθήκη ενός ρόλου σε έναν συγκεκριμένο χρήστη.

- **Διαδρομή:** `/user/role/add/{user_id}/{role_id}`
- **Μέθοδος:** GET
- **Αναμενόμενη Απόκριση:**
 - Προσθήκη του συγκεκριμένου ρόλου στον χρήστη.
 - Επιστροφή στη σελίδα λίστας χρηστών.

4.7 Διαγραφή Ρόλου από Χρήστη

Η ενότητα `/user/role/delete/{user_id}/{role_id}` επιτρέπει τη διαγραφή ενός ρόλου από έναν συγκεκριμένο χρήστη.

- **Διαδρομή:** `/user/role/delete/{user_id}/{role_id}`
- **Μέθοδος:** GET
- **Αναμενόμενη Απόκριση:**
 - Διαγραφή του συγκεκριμένου ρόλου από τον χρήστη.
 - Επιστροφή στη σελίδα λίστας χρηστών.

5. Ασφάλεια

5.1 Συνοπτική Εισαγωγή

Η κλάση `SecurityConfig` διαμορφώνει τις ρυθμίσεις ασφάλειας της εφαρμογής. Χρησιμοποιεί το Spring Security για τον έλεγχο της πρόσβασης στους πόρους της εφαρμογής.

5.2 Ρυθμίσεις Πρόσβασης

Η μέθοδος `filterChain` καθορίζει τις βασικές ρυθμίσεις πρόσβασης στους πόρους της εφαρμογής.

- **Διαδρομές:**

- `/`, `/home`: Προσβάσιμες χωρίς περιορισμούς.
- `/register`, `/saveUser`: Προσβάσιμες χωρίς περιορισμούς.
- `/course/**`: Προσβάσιμες μόνο από χρήστες που έχουν ρόλο "USER".
- Κάθε άλλη διαδρομή: Απαιτείται πιστοποίηση χρήστη.

5.3 Σελίδες Εισόδου και Εξόδου

Η εφαρμογή χρησιμοποιεί σελίδες σύνδεσης και αποσύνδεσης.

- Σελίδα Σύνδεσης: `/login`.
- Σελίδα Αποσύνδεσης: Προσβάσιμη από όλους με τη χρήση `/logout`.

5.4 Περιορισμοί Ρόλων

Κάποιες μέθοδοι των Controllers είναι περιορισμένοι μόνο για τον ρόλο διαχειριστή (ROLE_ADMIN)

Αυτό επιτυγχάνεται και με την χρήση του `@Secured("ADMIN")` annotation αλλά και με την χρήση του `sec:authorize="hasRole('ROLE_ADMIN')"` στον html κώδικα, ώστε κάποιες λειτουργίες/κουμπιά να μην εμφανίζονται σε απλούς χρήστες.

5.5 Ανεπαρκής Έλεγχος Πρόσβασης

Στην παρούσα υλοποίηση του συστήματος, παρατηρείται μια αδυναμία στον αποτελεσματικό έλεγχο πρόσβασης. Ένας χρήστης έχει τη δυνατότητα να υποβάλει αίτηση εξουσιοδοτήσεων από οποιονδήποτε πολίτη χωρίς να υπάρχει κατάλληλος έλεγχος ταυτότητας. Το ίδιο ισχύει και για τους ιατρούς, ένας χρήστης μπορεί να χειριστεί τις αιτήσεις των ιατρών χωρίς έλεγχο ταυτότητας. Αυτό το ελάττωμα θέτει σε κίνδυνο την ασφάλεια του συστήματος και την αυθεντικότητα των αιτήσεων.

6. Περιβάλλον Εκτέλεσης με Docker

Η εφαρμογή μας λειτουργεί μέσα σε ένα περιβάλλον Docker, που παρέχει ευελιξία και ευκολία στην ανάπτυξη.

Docker Container PostgreSQL:

- Η βάση δεδομένων της εφαρμογής τρέχει σε ένα container PostgreSQL. Με την παρακάτω εντολή, ξεκινάτε έναν PostgreSQL server:

```
docker run --name postgres --rm -e POSTGRES_PASSWORD=pass123 -e POSTGRES_USER=dbuser -e POSTGRES_DB=postgres -d --net=host
```

Πρόσβαση στην Εφαρμογή:

- Αφού η εφαρμογή εκτελείται, μπορείτε να την προσεγγίσετε στον περιηγητή σας με το URL <http://localhost:9090>.

7. Πηγές Κώδικα και Αλλαγές

Σε αυτό το κεφάλαιο, θα παρουσιάσουμε τις πηγές κώδικα που χρησιμοποιήσαμε κατά την ανάπτυξη της εφαρμογής μας, καθώς και τις τροποποιήσεις που κάναμε για να προσαρμόσουμε τον κώδικα στις ανάγκες μας.

• Κώδικας Από το Repository:

- Χρησιμοποιήσαμε κώδικα από το αποθετήριο <https://github.com/tsadimasteaching/ds-lab-2023> για να επιταχύνουμε την ανάπτυξη. Εξέτασαμε κυρίως τα παρακάτω κλαδιά:
 - **onetomany:** Χρησιμοποιήθηκε για τις αλληλεπιδράσεις μεταξύ του πολίτη και του ιατρού.

- **security:** Εξετάσαμε τις βασικές αρχές ασφαλείας και ενσωματώσαμε κάποιες από αυτές στην εφαρμογή μας.
- **main:** Ο τελικός κώδικας του κύριου κλαδιού για συγκεκριμένες βασικές λειτουργίες.

• Προσαρμογές και Τροποποιήσεις:

- Πραγματοποιήσαμε αλλαγές στον κώδικα που λάβαμε από το αποθετήριο για να ευθυγραμμίσουμε τις λειτουργίες με τις απαιτήσεις της δικής μας εφαρμογής. Αυτές οι τροποποιήσεις περιλαμβάνουν την προσθήκη, την αφαίρεση ή την τροποποίηση κομματιών του κώδικα για να καλύψουμε τις ανάγκες μας.

Η αναφορά αυτών των πηγών και οι προσαρμογές που κάναμε είναι σημαντικές για τη διαφάνεια και την κατανόηση του πώς δομήθηκε η εφαρμογή μας.