



Project HY-359

2021-2022

MedCrete

Dimitris Kourgiantakis csd4351
Theodoros Chalkidis csd4198

Στην παρακάτω αναφορά θα προσπαθήσουμε να συνοψίσουμε όλες τις λειτουργίες του website μας. Το site χωρίζεται σε 3 λειτουργικά κομμάτια (html/css, javascript, java servlets) καθώς και σε 4 γενικές κατηγορίες ανάλογα με την ιδιότητα του client (admin, user, doctor, guest).

Η άσκηση ήταν αρκετά απαιτητική διότι ζητούσε αρκετές λειτουργίες οι οποίες θα άλλαζαν εντελώς την εμφάνιση του front-end. Σαν κύριες τεχνολογίες χρησιμοποιήθηκαν οι εξής:

bootstrap 4.0/5.0, jquery, ajax, material bootstrap και leaflet για το front-end και servlets, gson, xampp, apache και my-sql-connector για την υλοποίηση του back-end.

Η αναφορά θα χωριστεί ανάλογα με την ιδιότητα του client και θα εξηγεί περίπου το τι έκανε το καθένα από τα τρία λειτουργικά κομμάτια για την υλοποίησή του.

Κεφάλαιο 1ο admin (σελ 2)

Κεφάλαιο 2ο doctor (σελ 3-4)

Κεφάλαιο 3ο user (σελ 5-6)

Κεφάλαιο 4ο guest (σελ 7)

Κεφάλαιο 5ο σχόλια υλοποίησης και επιμέρους σχολιασμός άσκησης (σελ 8)

Admin Panel

front-end/back-end

Το admin panel γίνεται access μέσα από το απλό login interface δίνοντας τα στοιχεία του admin. Έχει 2 λειτουργίες με το που συνδεθεί, την **edit all users** που μπορεί να διαγράψει όποιον user θέλει και την **certificate doctors** που κάνει approve to registration των γιατρών.

01 Html / css

Html / Css

Εμφανίζουμε ένα βασικό template που μετέπειτα το εμπλουτίζουμε με js.

02 Javascript

Javascript

Είναι υπεύθυνη για την δημιουργία και την λειτουργία των έξτρα λειτουργιών του admin. Κάνει τα εξής request, isLoggedIn το οποίο τσεκάρει αν είναι συνδεδεμένος ώστε να του επιτραπεί η πρόσβαση, certifyDoctor όπου πρώτα με άλλο request παίρνει όλους τους γιατρούς που υπάρχουν και δείχνει όσους δεν είναι certified και μετέπειτα βάζει ένα κουμπί δίπλα τους ώστε όταν πατηθεί να κληθεί το ApproveDoctors servlet. Το deleteUser που αφού πάρει όλους τους χρήστες της βάσης του προβάλλει με ένα κουμπί διαγραφής δίπλα από τον καθένα ώστε να μπορεί να τους διαγράψει άμα χρειαστεί και τέλος το Logout που καλεί το logout sevlet για την αποσύνδεση του admin από το website.

03 Servlets

Servlets

Χρησιμοποιούνται τα παρακάτω servlet:

Login/Logout: τα ίδια ακριβώς που είναι υπεύθυνα και για τον χρήστη (POST/GET)

AdminFields: επιστρέφει όλους τους γιατρούς και χρήστες (Post)

ApproveDoctors: δέχεται ένα username από γιατρό και τον κάνει approve (POST)

DeleteUser: δέχεται ένα username από οποιονδήποτε χρήστη (ελέγχεται να μην είναι ο admin) και τον διαγράφει (POST)

Το admin panel ήταν το πρώτο που υλοποιήθηκε καθώς ήταν και το πιο απλό. Επίσης έχει διαφορετική εικόνα από τους υπόλοιπους χρήστες και μπορεί κανείς να συνδεθεί σε αυτόν μόνο από το απλό login και όχι από των γιατρών.

Doctor Panel

front-end

Το doctor panel γίνεται access μέσα από το doctor login interface δίνοντας τα στοιχεία του doctor. Έχει 4 λειτουργίες με το που συνδεθεί, την **add new appointment**, την **view cancelled appointments**, την **view done appointments** και την **change settings**

01 Html / css

Html / Css

Εμφανίζουμε ένα βασικό template που μετέπειτα το εμπλουτίζουμε με js.

02 Javascript

Javascript

Είναι υπεύθυνη για την δημιουργία και την λειτουργία των έξτρα λειτουργιών του doctor. Κάνει τα εξής request, isLoggedIn το οποίο τσεκάρει αν είναι συνδεδεμένος ώστε να του επιτραπεί η πρόσβαση, addNewApp που στέλνει το νέο ραντεβού στην βάση, updateRendezvous που κάνει done ή cancel ένα ήδη υπάρχων ραντεβού και ανανεώνει τους πίνακες, sendNewTreatment που στέλνει ένα νέο treatment για την τελευταία εξέταση ενός χρήστη που έχει γίνει τουλάχιστον ένα ραντεβού μαζί του, drawCharts όπου δείχνει σε chart όλες τις εξετάσεις του χρήστη με τον οποίο έχει γίνει τουλάχιστον ένα ραντεβού done, sendMessage που στέλνει στην βάση το μήνυμα που θέλει ο γιατρός και ανανεώνει την συνομιλία τους με έναν χρήστη που έχει γίνει τουλάχιστον ένα ραντεβού μαζί του, τις addMessagePatient και addMessageDoctor που παίρνουν τα μηνύματα από τον server και δημιουργούν την συνομιλία μεταξύ γιατρού και χρήστη, την openChat η οποία ανοίγει το chat ενός γιατρού και ενός χρήστη που είχαν ένα τουλάχιστον επιτυχημένο ραντεβού, την openHistory που ανοίγει το ιστορικό εξετάσεων ενός χρήστη που έχει γίνει τουλάχιστον ένα ραντεβού μαζί του, και διάφορες ακόμα εσωτερικές συναρτήσεις για την αλλαγή του html/css και την εμφάνιση των δεδομένων που πρέπει σε κάθε περίπτωση. Τέλος όπως και ο χρήστης ένας γιατρός μπορεί να αλλάξει τα στοιχεία του και να κάνει logout.

Το doctor panel είχε αρκετή δουλειά στην javascript για να εμφανίζονται όμορφα τα δεδομένα δυναμικά και να στέλνονται σωστά τα δεδομένα στον σερβερ για την αποφυγή λαθών. Τεχνάσματα, όπως ότι σερβερ γυρνάει την ώρα στα μηνύματα σαν success ώστε να μην αποκλίνουν ούτε δευτερόλεπτο, χρησιμοποιήθηκαν για την τελειοποίηση του front end κομματιού.

Doctor Panel

Back-end

Το doctor panel γίνεται access μέσα από το doctor login interface δίνοντας τα στοιχεία του doctor. Έχει 4 λειτουργίες με το που συνδεθεί, την **add new appointment**, την **view cancelled appointments**, την **view done appointments** και την **change settings**

03 Servlets

Servlets

Χρησιμοποιούνται τα παρακάτω servlet:

Login/Logout: τα ίδια ακριβώς που είναι υπεύθυνα και για τον χρήστη (POST/GET)

CreateMessage: δημιουργεί μήνυμα και το βάζει στην βάση (Post)

CreateRendezvous: δημιουργεί ραντεβού και το βάζει στην βάση μετά από έλεγχο για σωστά πεδία και αν δεν έχει δοθεί θεραπεία για την ίδια εξέταση (POST)

CreateTreatment: δημιουργεί μία θεραπεία και το βάζει στην βάση μετά από έλεγχο για σωστά πεδία (POST)

GetMessage: επιστρέφει όλα τα μηνύματα από τον χρήστη και τον γιατρό που δόθηκαν (POST)

GetRendezvous: επιστρέφει όλα τα ραντεβού (free,selected,cancelled) του γιατρού που δόθηκε (POST)

UpdateRendezvous: ανανεώνει το ραντεβού σε done μόνο άμα έχει γίνει selected ή σε cancelled με βάση το id που δόθηκε (POST)

UpdateSettingsDoctor: ανανεώνει τα στοιχεία του γιατρού (POST)

UserBloodtest: επιστρέφει όλες τις εξετάσεις του χρήστη με το id που δόθηκε και όλες τις θεραπείες (POST)

Για το backend κομμάτι πολλούς ελέγχους τους κάναμε στο frontend για βελτιστοποίηση ταχύτητας και να μην έχουμε βαριά request ή άδεια request εξαιτίας απροσεξίας του χρήστη. Χρησιμοποιήθηκε επίσης η time για να έχουμε coherence στα μηνύματα ώστε να θεωρούμε σωστή ώρα αυτή που ανέθεσε ο σερβερ και όχι αυτή που στελνόντουσαν. Μικρή διαφορά στα millisecond είχαμε αλλά το θεωρήσαμε καλύτερη τακτική.

User Panel

front-end

Το user panel γίνεται access μέσα από το user login interface δίνοντας τα στοιχεία του user. Έχει πολλαπλές λειτουργίες με το που συνδεθεί, όπως το να μπορεί να κλείσει ραντεβού, να δει παλιές μετρήσεις, να δει θεραπείες (τελειωμένες ή και ενεργές), να αλλάξει τις ρυθμίσεις του κ.α.

01 Html / css

Html / Css

Εμφανίζουμε ένα βασικό template που μετέπειτα το εμπλουτίζουμε με js.

02 Javascript

Javascript

Είναι υπεύθυνη για την δημιουργία και την λειτουργία των έξτρα λειτουργιών του user. Κάνει πάρα πολλά request (1 με 2 για κάθε λειτουργία του).

Λειτουργίες :

- BMI : request με τα κιλά και το ύψος του για την εμφάνιση του ύψους
- Show All Randevouz: βλέπει όλα τα ραντεβού του και μπορεί να επικοινωνήσει με τον γιατρό μέσω μηνύματος.
- Add new blood test: προσθέτει νέες εξετάσεις και βλέπει όλες του τις παλιές εξετάσεις
- Show bloodtest history: βλέπει όλες τις παλιές του εξετάσεις ανάμεσα σε δύο ημερομηνίες οπτικοποιημένες για σύγκριση καθώς και όλες τις θεραπείες που σχετίζονται με αυτές τις εξετάσεις (ακόμα και αν τελείωσαν ή αν είναι ακόμα ενεργές)
- Show active treatments: βλέπει όλες τις ενεργές του εξετάσεις
- Doctors: μπορεί να δει όλους τους γιατρούς και όλα τα ελεύθερα τους ραντεβού και μπορεί να κλείσει ραντεβού. Τους γιατρούς μπορεί να τους δει με βάση την απόσταση, την ώρα για να φτάσει ή την χαμηλότερη τιμή.
- Ακόμα υποστηρίζονται τα ίδια πράγματα για τα login/logout όπως και στους γιατρούς
- Έρχεται ειδοποίηση αυτόματα στο login άμα έχει ραντεβού σε λιγότερο από 4 ώρες σαν pop up notification

Πολλές από τις λειτουργίες για να τρέξουν σωστά κάνουν πάνω από ένα request όπως πχ η doctors η οποία μετά που θα γυρίσει τους γιατρούς και τα ραντεβού πρέπει να επιτρέψει τον χρήστη να μπορεί να κλείσει και ένα ραντεβού. Ακόμα στο front end γίνονται οι περισσότεροι έλεγχοι για τις τιμές των πεδίων γιατί είναι πιο άμεσο και εύκολο.

User Panel

Back-end

Servlets

Χρησιμοποιούνται τα παρακάτω servlet:

- **Login/Logout:** τα ίδια ακριβώς που είναι υπεύθυνα και για τον χρήστη (POST/GET)
- **CreateMessage:** δημιουργεί μήνυμα και το βάζει στην βάση (Post)
- **CreateRandevouz:** δημιουργεί ραντεβού και το βάζει στην βάση μετά από έλεγχο για σωστά πεδία και αν δεν έχει δοθεί θεραπεία για την ίδια εξέταση (POST)
- **ActiveRandevouz:** επιστρέφει μήνυμα αν ο χρήστης έχει ραντεβού σε 4 ώρες
- **GetMessage:** επιστρέφει όλα τα μηνύματα από τον χρήστη και τον γιατρό που δόθηκαν (POST)
- **GetRandevouzUser:** επιστρέφει όλα τα ραντεβού του χρήστη που δόθηκε (POST)
- **UpdateRandevouz:** ανανεώνει το ραντεβού σε done μόνο άμα έχει γίνει selected ή σε cancelled με βάση το id που δόθηκε (POST)
- **UpdateSettings:** ανανεώνει τα στοιχεία του χρήστη (POST)
- **UserBloodtest:** επιστρέφει όλες τις εξετάσεις του χρήστη με το id που δόθηκε και όλες τις θεραπείες (POST)
- **BloodTestMeasure:** επιστρέφει όλες τις μετρήσεις του ραντεβού του χρήστη με το id που δόθηκε (POST)
- **GetActiveTreatment:** επιστρέφει όλες τις ενεργές θεραπείες του χρήστη
- **GetAllDoctors:** επιστρέφει όλους τους γιατρούς
- **GetBloodTest:** επιστρέφει όλα τα bloodtest ανάμεσα στις ημερομηνίες που δώθηκαν
- **GetRandevouzPrice:** επιστρέφει doctor id με το χαμηλότερο τους price
- **NewBloodTest:** δημιουργεί ένα νέο bloodtest
- **UpdateRandevouzUser:** ανανεώνει το ραντεβού σε selected και βάζει τα στοιχεία του χρήστη ώστε να μπορεί να τα δει ο γιατρός

Guest Panel

front-end/back-end

Το guest panel γίνεται access μέσα από το από το κουμπί login as a guest στην αρχική σελίδα και δεν μπορεί να κάνει και πολλά πέρα από το να δει τους γιατρούς και κάποιες χρήσιμες πληροφορίες.

01 Html / css

Html / Css

Εμφανίζουμε ένα βασικό template που μετέπειτα το εμπλουτίζουμε με js.

02 Javascript

Javascript

Είναι υπεύθυνη για την δημιουργία και την λειτουργία των έξτρα λειτουργιών του admin. Ο guest μπορεί απλά να δει τους γιατρούς στον χάρτη και κάποιες χρήσιμες πληροφορίες. Χρησιμοποιήθηκε η βιβλιοθήκη leaflet για τους χάρτες.

03 Servlets

Servlets

Μόνο οι getAllDoctors χρησιμοποιείται

Project HY-359

2021-2022

MedCrete

Dimitris Kourgiantakis csd4351
Theodoros Chalkidis csd4198

Conclusion:

Το όλο πρότζεκτ ήταν πολύ απαιτητικό και χρειάστηκε πάρα πολύς χρόνος για την κατανόηση και υλοποίηση των ερωτημάτων. Τον μεγαλύτερο όγκο τον έπιανε το front end (html 29.2% , css 26,7%, javascript 19,2%) σε αντίθεση με το back-end που έγινε σε τζαβα και έπιανε μόνο 24,3%. Το πιο δύσκολο κομμάτι ήταν η δυναμική δημιουργία της html ώστε να εξυπηρετεί όλες τις ανάγκες. Ίσως να ήταν καλύτερο να γίνει πρότζεκτ για τρία άτομα για να είναι εφικτό το 100% σε λιγότερες μέρες καθώς συμπίπτει και με πρότζεκτ από άλλα μαθήματα. Επίσης το api για τους χάρτες είναι αρκετά δύσχρηστο και μας πήρε αρκετή ώρα.



