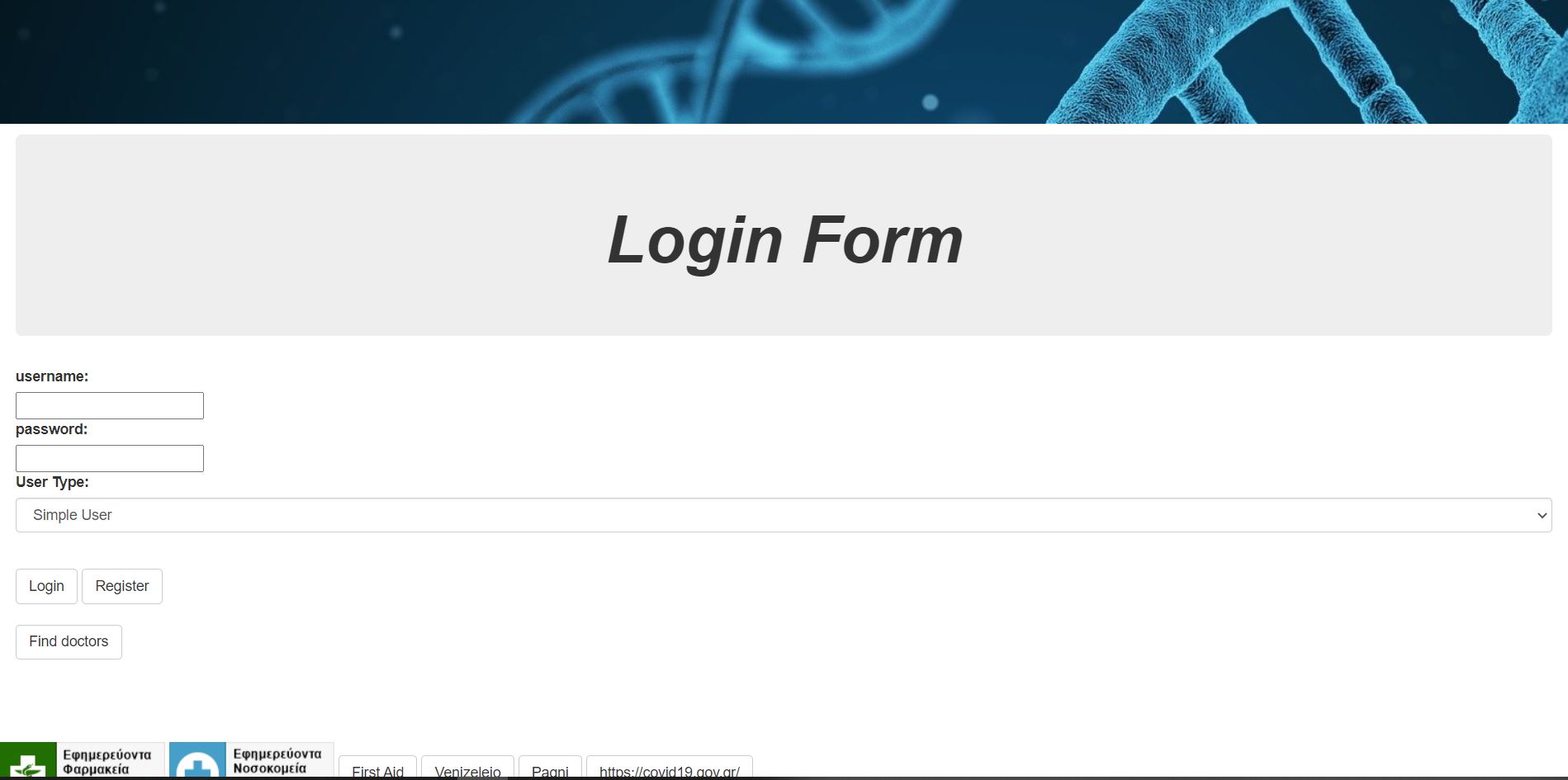
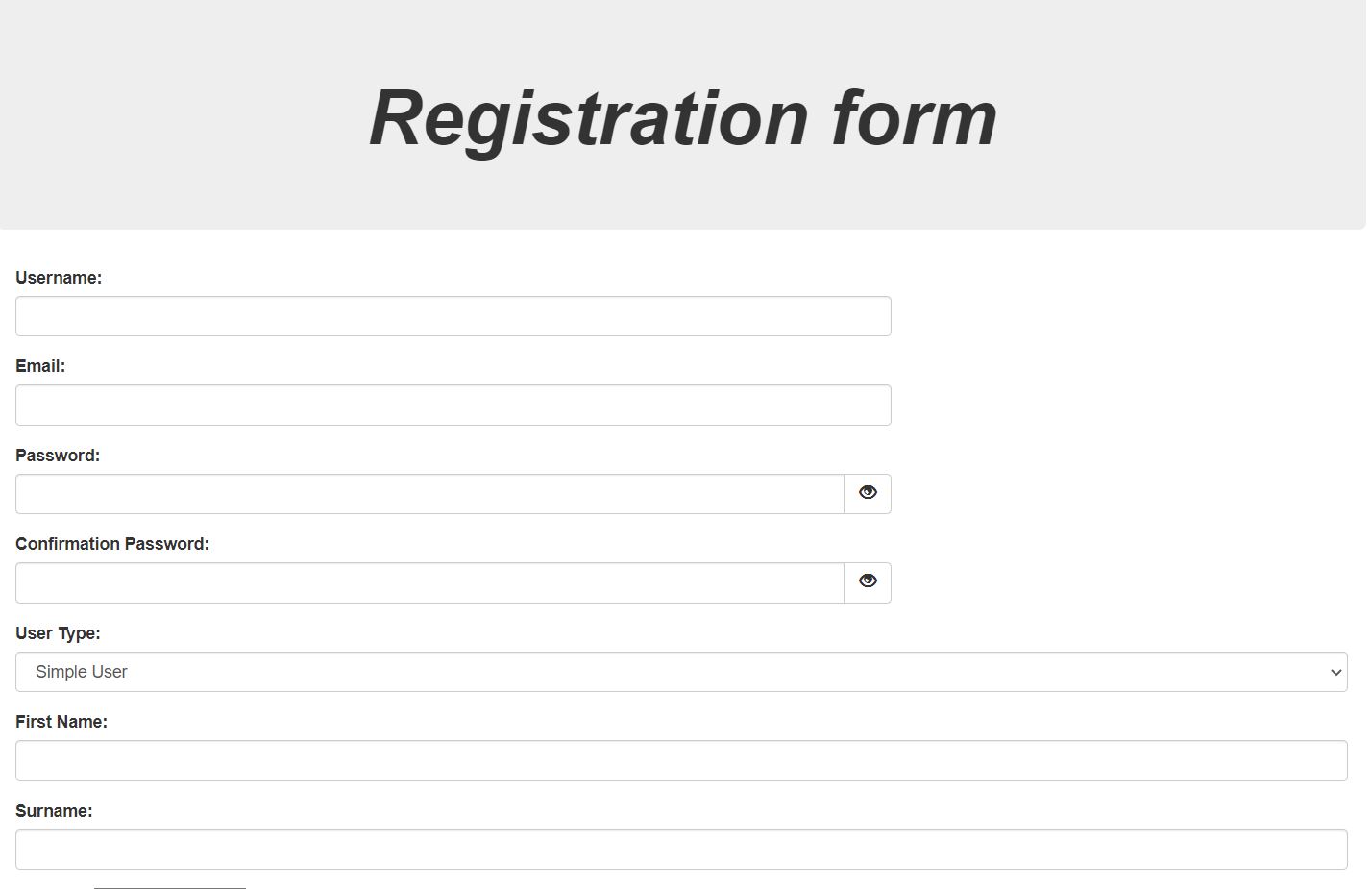
Κων/νος Ζιούτος 3946 [csd3946@csd.uoc.gr](mailto:csd3946@csd.uoc.gr)

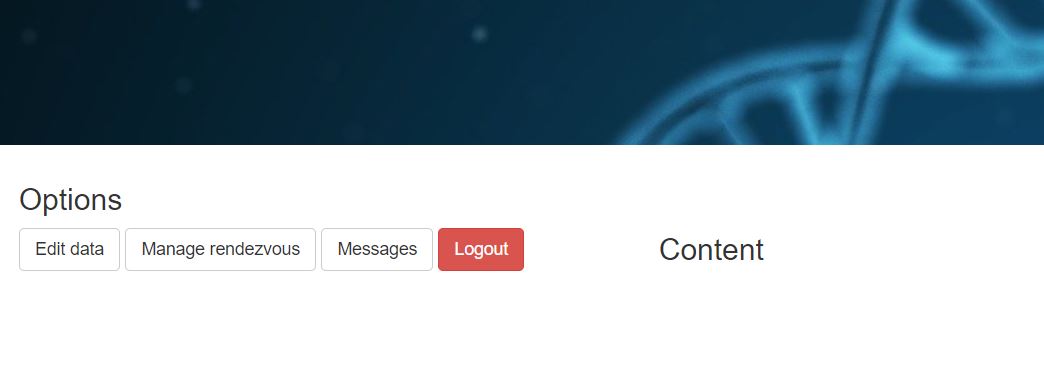
**Οθόνες συστήματος**

Αρχικά έχουμε την οθόνη που ο χρήστης κάνει login. Από εκεί μπορεί να πάει στη σελίδα που θα κάνει εγγραφή ως νέος χρήστης(ιδια λειτουργια με τη τριτη ασκηση).Παράλληλα έχουμε στη σελίδα του login τις λειτουργίες του επισκέπτη/μη συνδεδεμένου χρήστη που παρέχουν γενικές πληροφορίες. Χρησιμοποιήσαμε για τις οθόνες html και css, ενώ για να γίνει το login και το register χρησιμοποίουμε servlets που θα αναφέρουμε παρακάτω.Όταν γίνεται login εμφανίζονται τα πεδία που μας ενδιαφέρουν ανάλογα αν είναι απλός χρήστης γιατρός ή διαχειριστής

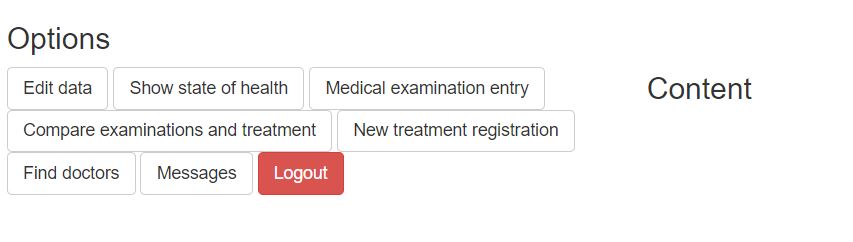




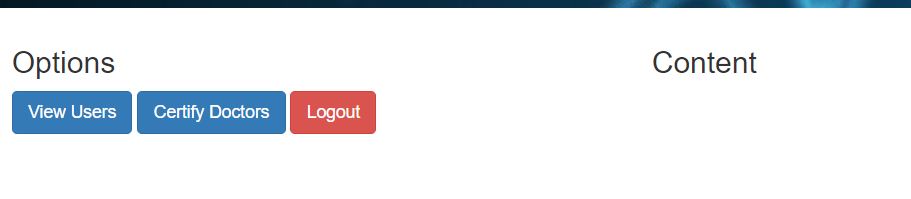
**Γιατρός**



**Απλός Χρήστης**

****

**Διαχειριστής**

****

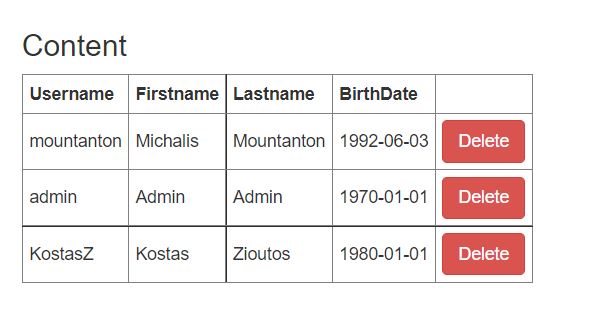
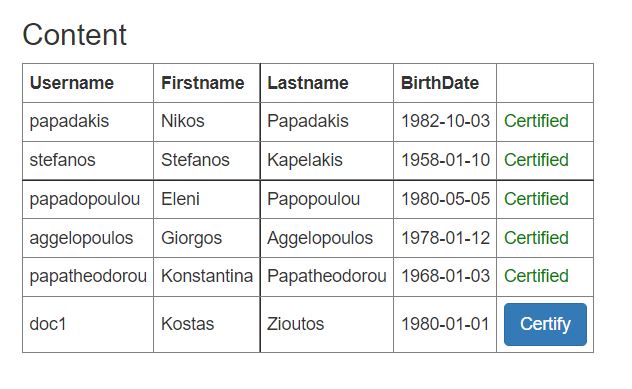
**Λειτουργίες Επισκέπτη/ Μη συνδεδεμένου Χρήστη**

Για τις πληροφορίες κάνουμε στο footer χρησιμοποίουμε κουμπια hyperlinks για τις σελίδες που θέλουμε. Ενώ για το να βρεί για γιατρούς καλούμε με onclick μια συνάρτηση από τη javascript την find\_docs η οποία κανει μια GET στο servlet Doctors που έχουμε φτίαξει από εκεί παίρνουμε με με json από το table των γιατρών της βάσης τους γιατρους σε μορφή array list. Μετά με άλλη συνάρτηση στη javascript τυπώνουμε στην οθόνη τα αποτελέσματα μας σε πίνακα.



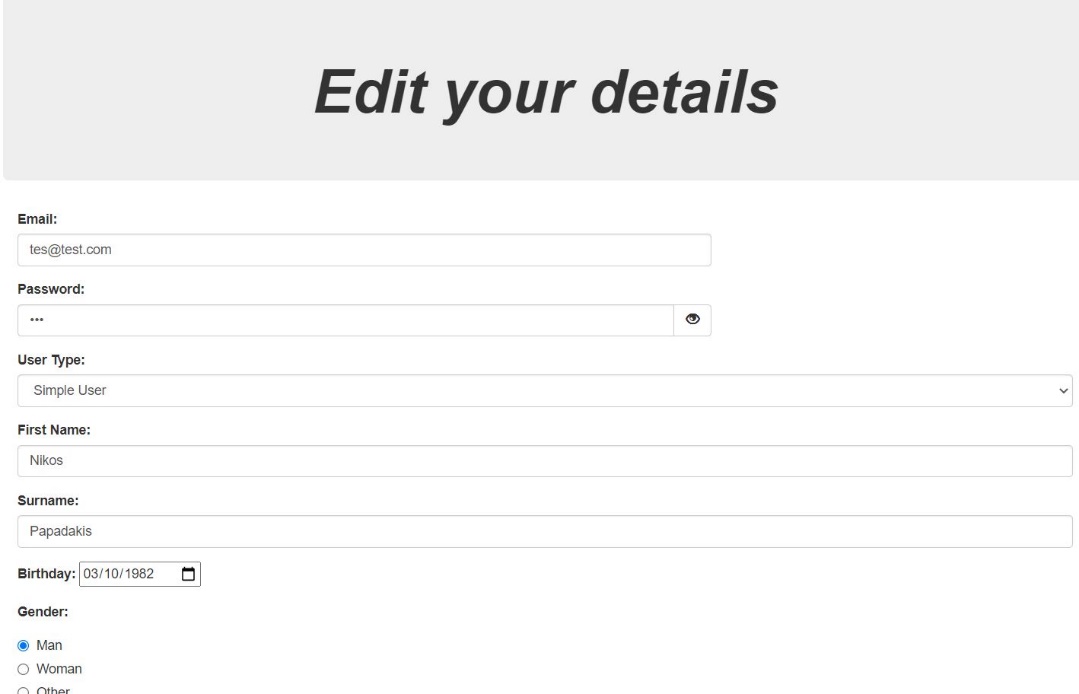
**Λειτουργίες Administrator**

Όταν δώσουμε τα στοιχεία του admin και επιλέξουμε να συνδεθούμε σαν admin καλουμε τη συνάρτηση login από τη javascript και αυτή κάνει ένα POST request στο servlet LoginUser που υλοποιήσαμε. Μέσα στη doPost κάνουμε έλεγχο τι είδος είναι ο χρήστης που συνδέθηκε(γιατρος,απλος,διαχειριστής) και έτσι εξασφαλίζουμε το διαχωρισμο ανάμεσα στυους χρήστες(με τα διαφορετικά attributes που παίρνει ο κάθε χρήστης κλπ).Όταν συνδεθούμε έχουμε τώρα τη δυνατότητα να προβάλουμε τους χρήστες έχοντας τη δυνατότητα να τους διαγράψουμε αλλά και να πιστοποιούμε γιατρούς. Όταν πατάμε την εκάστοτε επιλογή εμφανίζεται το περιεχόμενο της και κρύβεται το άλλο περιεχόμενο(αν υπάρχει).Ταυτόχρονα το πάτημα του κουμπιού στέλνει ένα GET στο servlet AdminAction μαζί με ένα query που μέσω αυτού θα καταλαβαίνουμε τη ενέργεια κάνει ο διαχειριστής.Στο servlet λοιπόν στη doGET ξεχωρίζουμε τη κάθε ενέργεια, για να στείλουμε πίσω τους γιατρούς ή τους χρήστες χρησιμοποιούμε json. Eνώ όταν θέλουμε να κάνουμε certify ένα γιατρό ή να διαγράψουμε ένα χρήστη καλούμε από το πίνακα των γιατρών ή του χρήστη αντίστοιχα τις certifyDoctor και deleteSimpleUser αντιστοιχα που κάνουν στη βάση αυτές τις ενέργειες.Και τις δύο τις υλοποιήσαμε εμείς και δεν υπήρχαν στο κώδικα της βάσης. Τέλος έχουμε το logout που δεν είναι διαφορετικό από όσα έχουμε δει στο μάθημα και γίνει με ένα POST



**Λειτουργίες Ιατρού**

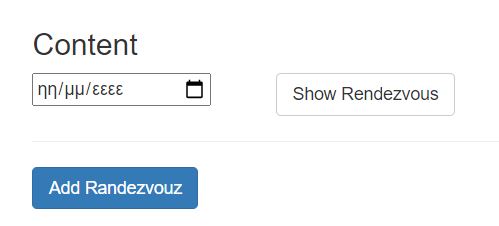
Όταν τώρα γίνεται login ένας γιατρός με την ίδια διαδικασία με πρίν βλέπει στο πάνελ του τρια κουμπια,να επεξεργαστεί τα δεδομένα του,να διαχειριστεί ραντεβού και να ανταλλάξει μηνύματα.Αν επιλέξω το πρώτο μας πάει σε μια ξεχωριστή σελίδα που μας δίνει τη δυνατότητα να επεξεργαστούμε τα στοιχεία μας και να αποθηκεύσουμε τα ανανεωμένα δεδομένα.





Μόλις πατηθεί το save πάμε στη javascript μια συνάρτηση saveDetails που στέλνει ένα POST στο servlet EditDetails που εκεί ξεχωρίζει τις περιπτώσεις για γιατρό ή απλο χρήστη και ανανεώνει μεσω json τους πίνακες από τους γιατρούς (στη προκειμένη περίπτώση) ή απλό χρήστη.

Όταν πατάμε στο manage rendezvous έχουμε ημερολόγιο και από εκεί παίρνοντας μια συγκεκριμένη ημερομηνία χτυπαέι με GET η fetchRendezvous στο servlet Rendezvous εκέι με το action παώ στην sendRandevouz που μου επιστρέφει μέσω json από το κώδικα της βασης και συγκεκριμένα η databaseToRandevouz που εμείς υλοποίησαμε και με το response εμφανίζουμε με της showDayRandevouz σε μορφή πίνακα τα ραντεβού της συγκεκριμένης ημέρας και μαζι της πληροφρορίες του,την κατάσταση του(Done κλπ) και τις επιλογές να δει τις εξετάσεις και να στείλεί μήνυμα. Aν δεν είναι Done και είναι selected μπορει να τα κάνει cancel ή done και αυτό το κάνουν οι cancelRandevouz κι doneRandevouz αντίστοιχα που και οι δυο στέλνουν ένα GET στο servlet των ραντεβού όπου με αντιστοιχη διαδικασία με παραπάνω ανανεώνει μέσω της sendRandevouz.





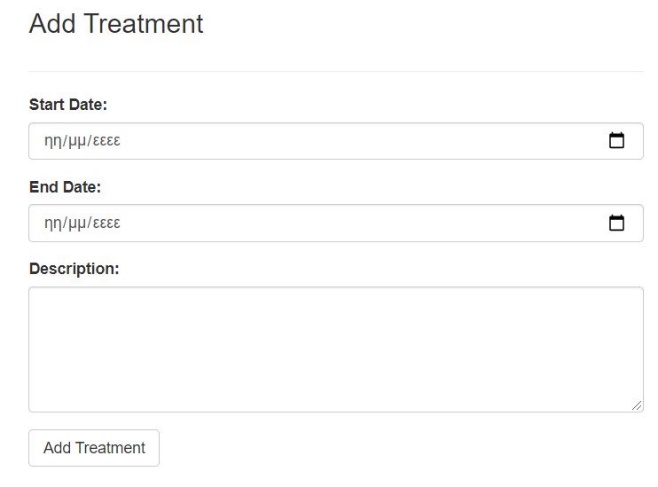
Αν θέλει να δει τα blood test όταν κάνει click τοτε η viewBloodTests με ένα GET στο servlet BloodTests που αυτή εκεί παίρνει το action getTests και παίρνοντας το id του user παει στο πίνακα των user και παίρνοντας το αμκα με την getAmka που φτιάξαμε εμεις το χρησιμοποιεί για να βρεί την arraylist από τα bloodtest.Kαι μετά με το response τα παρουσιάζουμε.Χρησιμοποιόντας επίσης και τα google charts και με την drawVisualization τα κάνουμε και visualize.



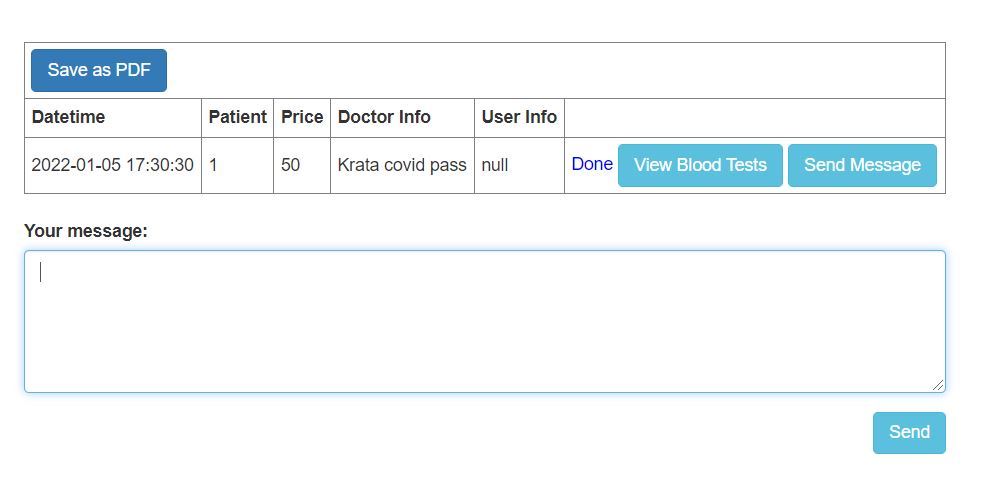
Tώρα έχουμε τη θεραπεία που μπορούμε και να δούμε και να προσθέσουμε.Για να τη δούμε καλούμε με το κουμπί την viewTreatment που κάνει GET στο servlet των treatments που σε αυτό πάει στη βάση και αν υπάρχει το επιστρέφει αλλιώς εμφανίζει μήνυμα. Στη συνέχεια με την showTreatment εμφανίζουμε το περιεχόμενο.



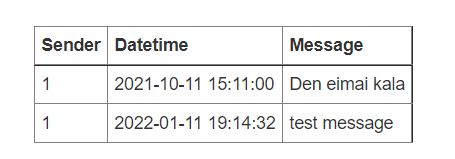
Τώρα για να προσθέσουμε θεραπεία χρησιμοποιούμε την addTreatment που κάνει POST στο servlet. Όπου εκεί κανουμε τη μετατροπη να πάρουμε το json από το AJAX,μετα κοιτάμε αν υπάρχει ήδη treatment αλλιώς παίρνουμε τον simple user όπως είπαμε πριν με την getSimpleUserByAmka(που έχουμε φτιάξει εμεις) και προσθέτουμε θεραπεία για αυτόν.



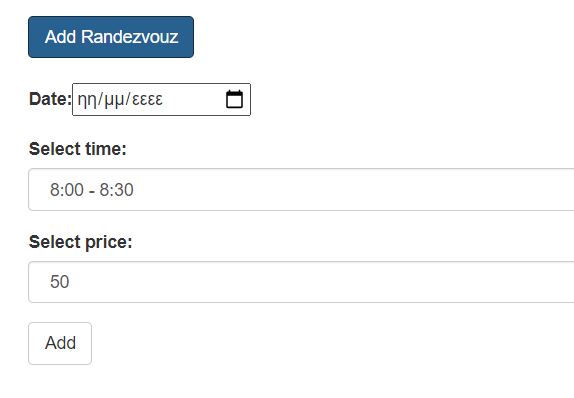
Για να στείλουμε μηνύματα χρησιμοποιούμε την sendMessageToUser που κανει POST στο servlet Messages όπου από εκεί αποθηκευεται το μήνυμα στη βάση που έχουμε γράψει με ημερομηνία κλπ.



Eνώ για να δούμε τα μηνύματα χρησιμοποιούμε την showMessages(usertype). Tο usertype τωρα είναι το user id αλλά φτιάχτηκε έτσι ώστε να χρησιμοποιηθεί για να δεί και ο χρήστης τα μηνύματα του γιατρου.Η οποιά κάνει GET για να πάρει τα μηνύματα και τα εμφανίζει σε πίνακα.

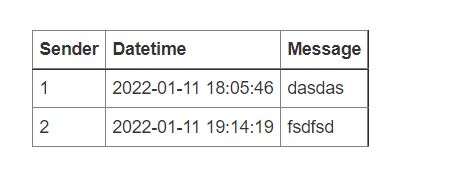


Όσο τώρα για να προσθέτουμε ραντεβού μολις πατήσουμε το κουμπί εμφανίζουμε μια φορμα ώστε να δημιουργήσουμε το ραντεβού την κατάλληλή ημερομηνία και ώρα.Όταν κάνουμε add τρέχουμε την addRandevouz που κάνει GET στο σερβλετ των ραντεβού όπου εκεί κάνουμε τους καταλληλους ελέγχους και αν είναι όλα καλα δημιουργούμε ένα ραντεβού καινούργιο με την createNewRandevouz.



**Λειτουργίες Εγγεγραμμένου Χρήστη**

Όταν κάνει login κάποιος σαν χρήστης βλέπει σαν επιλογές όλες τις λειτουργίες που μπορεί να κάνει. Εμείς λόγω φόρτου εργασίας με διάφορα πρότζεκτ και την εξεταστική που έρχεται/τρέχει έχουμε υλοποιήσει το edit data που επεξεργάζεται τα δεδομένα του χρήστη όπως είπαμε παραπάνω και τον τρόπο που αυτό γίνεται.Και επίσης έχουμε να μπορεί να βλέπει τα μηνύματα που του στέλνει ο γιατρός αυτό υλοποιήθηκε όπως και παραπάνω απλά εδώ αλλάξαμε το usertype που αναφέραμε παραπάνω.



Αλλαγές/προσθηκες functions στο κώδικα της βάσης

* EditBloodTestTable: public ArrayList<BloodTest> databaseToBloodTest(String amka) //Για να παιρνουμε τα αποτελέσματα σε array list
* EditDoctorTable: public boolean itemExists(String item, DoctorItem itemType)
* EditDoctorTable : public String databaseToJSON(String username) //για να παίρνουμε μόνο βάση του ονοματος
* EditDoctorTable : public void certifyDoctor(int doctorId)
* EditDoctorTable : public void updateDoctor(Doctor doctor) //για να γινει με παράμετρο τον γιατρό
* EditMessageTable: public ArrayList<Message> databaseToUserMessages(int userId)
* EditMessageTable: public ArrayList<Message> databaseToDoctorMessages(int userId)
* public void updateRandevouzStatus(int randevouzID, String status) //αλλαγη στατους ραντεβου
* EditRandevouzTable: public ArrayList<Randevouz> databaseToRandevouz(String date, int doctorId)
* EditRandevouzTable: public Randevouz databaseToRandevouz(String datetime)
* EditSimpleUserTable: public String databaseUserToJSON(String username)
* EditSimpleUserTable: public SimpleUser getSimpleUserByAmka(String amka) //τον παιρνει με αμκα
* EditSimpleUserTable: public SimpleUser databaseToSimpleUser(String username)
* EditSimpleUserTable: public ArrayList<SimpleUser> getUsers()
* EditSimpleUserTable: public boolean itemExists(String item, UserItem itemType)
* EditSimpleUserTable: public void deleteSimpleUser(int userId)
* EditTreatmentTable public Treatment getBloodtestTreatment(int bloodtestId)// παιρνει τη θεραπεια με βαση τις εξετασεις