

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής



Εργασία Μαθήματος «Προγραμματισμός στο διαδίκτυο και στον
παγκόσμιο ιστό»

«αριθμός άσκησης»	Τίτλος άσκησης
Όνομα φοιτητή – Αρ. Μητρώου (όλων σε περίπτωση ομαδικής εργασίας)	Στυλιανή Καλογήρου, Π18181
	Χάρης Πολυκάρπου, Π16161
Ημερομηνία παράδοσης	14/07/2021



Εκφώνηση της άσκησης

Δημιουργία 3-tier εφαρμογής για την διαχείριση ραντεβού ιατρικών εξετάσεων

Στόχοι εργασίας: Ολοκλήρωση λειτουργικότητας 3-tier εφαρμογής, ολοκλήρωση server-side τεχνολογιών

(*servlets και προαιρετικά jsp*), επικοινωνία με βάση δεδομένων, ολοκλήρωση λειτουργιών.

Στην τελική εργασία του μαθήματος θα επεκτείνετε την 2η Άσκηση ώστε να ολοκληρώσετε την εφαρμογή

τριών επιπέδων (3-tier), η οποία θα υλοποιεί όλες τις λειτουργίες (μεθόδους) που ορίσατε στην 1η Άσκηση

(με τις πιθανές αλλαγές που έγιναν).

Αναλυτικά Βήματα:

1. Επέκταση web project προηγούμενης άσκησης

1.1. Στην τελική εργασία θα επεκτείνετε τη λειτουργικότητα του web project που δημιουργήσατε στην

προηγούμενη άσκηση και θα υλοποιήσετε όλη την ζητούμενη λειτουργικότητα για κάθε

κατηγορία χρηστών.

2. Δημιουργία διαδικτυακής διεπαφής

2.1. Για την είσοδο των χρηστών στο σύστημα θα υλοποιήσετε **μηχανισμό login** με username και

password. Το password θα αποθηκεύεται σε κρυπτογραφημένη (hashed+salted) μορφή. Από την

αρχική σελίδα οι διάφοροι χρήστες θα μπορούν να έχουν πρόσβαση στις λειτουργίες τους.

2.2. Σε αυτό το βήμα, θα υλοποιήσετε τις διαδικτυακές διεπαφές (html ή jsp σελίδες) που θα

χρησιμοποιούν οι χρήστες όλων των κατηγοριών (Ασθενείς, Ιατροί, Διαχειριστές) για να

αλληλεπιδρούν με την εφαρμογή και να χρησιμοποιούν τις αντίστοιχες μεθόδους που απαιτούνται.

2.2.1. Θα υπάρχει ένα κεντρικό μενού σε μία index.html (ή index.jsp) σελίδα, η οποία θα είναι η

αρχική σελίδα για όλους τους χρήστες. Μετά το login θα προβάλλεται το μενού λειτουργιών

κάθε χρήστη ανάλογα με την κατηγορία στην οποία ανήκει.

2.2.2. Λειτουργίες Ασθενών (Patient): Οι Ασθενείς θα μπορούν να εκτελούν κατ ελάχιστο τις



λειτουργίες: προβολή ιστορικού προηγούμενων ραντεβού, προβολή διαθέσιμων κενών για κλείσιμο ραντεβού με έναν γιατρό κάποιας ειδικότητας, κλείσιμο ραντεβού ακύρωση ραντεβού (σε περίπτωση που το ραντεβού είναι προγραμματισμένο τουλάχιστον 3 ημέρες μετά).

2.2.3. Λειτουργίες Ιατρών (Doctor): Οι Ιατροί θα μπορούν να εκτελούν κατ ελάχιστο τις λειτουργίες: καταχώρηση διαθεσιμότητας για ραντεβού (ανά μήνα), προβολή πίνακα ραντεβού, ακύρωση ραντεβού (σε περίπτωση που είναι τουλάχιστον 3 ημέρες μετά).

2.2.4. Λειτουργίες Διαχειριστή (Administrator). Οι Διαχειριστές θα μπορούν να εκτελούν κατ

ελάχιστο τις λειτουργίες: εισαγωγή νέου Ιατρού και χρήστη, διαγραφή Ιατρού.

2.3. Η εφαρμογή θα υποστηρίζει **διαχείριση συνόδου (session management)** από τη στιγμή που ο

χρήστης συνδέεται, μέχρι την αποσύνδεσή του από την εφαρμογή. Κατά την αποσύνδεση του

χρήστη θα πρέπει να διαγράφεται το session.

3. Υλοποίηση επιπέδου Δεδομένων και σύνδεση εφαρμογής με τη βάση

3.1. Όλα τα δεδομένα θα αποθηκεύονται σε βάση δεδομένων, την οποία έχετε σχεδιάσει από την 2η

Άσκηση (π.χ. μέσω mysql + mysql workbench ή postgresql ή άλλης αντίστοιχης τεχνολογίας).

Μπορείτε να προβείτε σε όποιες τροποποιήσεις θεωρείτε απαραίτητες. Προσθέστε δοκιμαστικά δεδομένα στη βάση.

3.2. Διαμορφώστε κατάλληλα το project σας ώστε να συνδέσετε τη Βάση Δεδομένων που έχετε

δημιουργήσει με τον application server σας, ως μία 3-tier εφαρμογή (σύνδεση του application server

με τη Βάση Δεδομένων και της εφαρμογής σας μέσω του application server – μπορείτε να βρείτε

αντίστοιχο παράδειγμα στα παραδείγματα κώδικα που περιλαμβάνονται στη σελίδα του μαθήματος).

4. Υλοποίηση επιπέδου επεξεργασίας (servlet)

4.1. Διαμορφώστε κατάλληλα το project σας ώστε να επικοινωνεί με τον application server της

επιλογής σας (στα java παραδείγματα έχουμε χρησιμοποιήσει apache tomcat).

4.2. Υλοποιήστε όλες τις λειτουργίες που προσφέρει η εφαρμογή σας

χρησιμοποιώντας τεχνολογία

servlet. Δημιουργήστε ένα ή περισσότερα servlet τα οποία θα δέχονται είσοδο από το επίπεδο



διεπαφής (html ή jsp σελίδες και φόρμες), θα αναζητούν στη βάση δεδομένων τα στοιχεία που απαιτούνται ότι απαιτείται και θα επιστρέφουν το αποτέλεσμα στον εκάστοτε χρήστη ως δυναμική html σελίδα.

4.3. Προαιρετικά, μπορείτε να χρησιμοποιήσετε τεχνολογία jsp για τη δημιουργία και την διαμόρφωση των ιστοσελίδων.



ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	Γενική Περιγραφή της λύσης.....	6
2	Βάση Δεδομένων.....	8
2.1	Μοντέλο Οντοτήτων-Σχέσεων	8
2.2	Επεξήγηση Πινάκων	8
3	Κώδικας προγράμματος	8
3.1	Κώδικας του αρχείου Login.jsp και Login_Request.jsp.....	8
3.1	Κώδικας του αρχείου logout.jsp	11
3.2	Κώδικας του αρχείου Doctors.jsp	12
3.3	Κώδικας των κλάσεων (Admins, Information, Members)	12
3.4	Κώδικας του αρχείου Panel_Users_View.jsp.....	12
3.5	Κώδικας του αρχείου Panel_Add_Appointment.jsp και.....	13
3.6	Κώδικας του αρχείου Contact.jsp	14
3.7	Κώδικας του αρχείου HosplInfo.jsp	14
3.8	Κώδικας του αρχείου Doctors.jsp	16
4	Βιβλιογραφικές Πηγές.....	16
4.1	IntelliJ IDEA 2021.1.....	16
4.2	AdoptOpenJDK with Hotspot 8.212	16
4.3	PGAdmin 4.8.....	17
4.4	PostgreSQL 10.5-1	17
4.5	Tomcat 9.0.35.....	17
4.6	PostgreSQL-42.2.5 (Driver).....	17
4.7	SERVLETS και ΣΕΛΙΔΕΣ ΔΙΑΚΟΜΙΣΤΗ JAVA – ΤΕΧΝΟΛΟΓΙΕΣ ΠΥΡΗΝΑ 2 ^Η ΑΜΕΡΙΚΑΝΙΚΗ ΕΚΔΟΣΗ – MARTY HALL & LARRY BROWN	17
4.8	Συστήματα Βάσεων Δεδομένων 6 ^Η Έκδοση – Abraham Silberschatz, Henry F. Korth & S. Sudarshan.....	17
4.9	Menu https://www.w3schools.com/css/css_navbar.asp	17
4.10	Ιδέες για τον τρόπο που θα γίνει το registration/login system https://www.javatpoint.com/registration-form-in-jsp	17



1 Γενική Περιγραφή της λύσης

Στο συγκεκριμένο στάδιο της εργασίας δημιουργήθηκαν οι κλάσεις Users, Patients, Doctors, Admins. Αντίστοιχα δημιουργήθηκαν τα WEB αρχεία index.jsp (αρχική), default.jsp, Login.jsp, LoginRequest.jsp, Register.jsp, RegisterRequest.jsp, Profile.jsp, Profile_Request.jsp.

Συνοπτικά, το πρόγραμμα παρουσιάζει στον επισκέπτη τις επιλογές Home, Doctors, About Hospital, Register και Login, με αποτέλεσμα να μπορεί να κάνει εγγραφή ή σύνδεση στην σελίδα σαν χρήστης, Διαχειριστής και σαν Διαχειριστής περιεχομένου.

Επίσης το αρχείο default.jsp περιέχει το menu που θα ακολουθεί τον χρήστη σε όλες τις σελίδες και ανάλογα με τις άδειες που έχει θα του εμφανίζει τις κατάλληλες επιλογές.

Επιπρόσθετα, η υλοποιημένες κλάσεις που έχουν σκοπό την οργάνωση του κώδικα (prepareStatements, queries, κτλ) καλούνται αντίστοιχα εκεί που χρειάζεται.

Οι υπόλοιπες JSP σελίδες ακολουθούν την ίδια δομή κώδικα περίπου με μόνο αλλαγές στα queries, στους ελέγχους ανάλογα και στο CSS.



2 Βάση Δεδομένων

2.1 Μοντέλο Οντοτήτων-Σχέσεων

αρχείο **database1.png**.

2.2 Επεξήγηση Πινάκων

Με την εντολή **CREATE TABLE** ορίσαμε τους **πίνακες** μας με τα ονόματα **«users» «appointments»**, **<doctors>** Ο πίνακας **«users»** περιέχει επτά (7) στήλες: **user_id, username, password, full_name, email, telephone, role**.

- Ο πίνακας **«appointments»** περιέχει τέσσερις (4) στήλες: **doct_id, id, description, date**.
- Ο πίνακας **«doctors»** περιέχει 4 στήλες: **doct_id, username, doct_name, identity**
- Οι στήλες **users.id, doct_id, appid** έχουν τους αντίστοιχους τύπους δεδομένων (integer) δηλαδή είναι ένας ακέραιος αριθμός ο οποίος αυξάνεται κατά ένα κάθε φορά που προθέτεται μια νέα εγγραφή.
- Όσες στήλες έχουν τον περιορισμό **NOT NULL**, δεν μπορούν να πάρουν κενό όρισμα και πρέπει να έχουν αναγκαστικά μια.
- Τα πρωτεύων κλειδιά (**PRIMARY KEYS**) δηλαδή **users.id, Doctors.id** είναι οι στήλες στις οποίες απαγορεύεται οποιεσδήποτε δύο εγγραφές στην ίδια στήλη να έχουν ταυτόχρονα την ίδια τιμή.
- Τέλος ο περιορισμός **UNIQUE** εξασφαλίζει ότι όλες οι τιμές των στηλών **users.email, users.username** θα είναι διαφορετικές.

3 Κώδικας προγράμματος

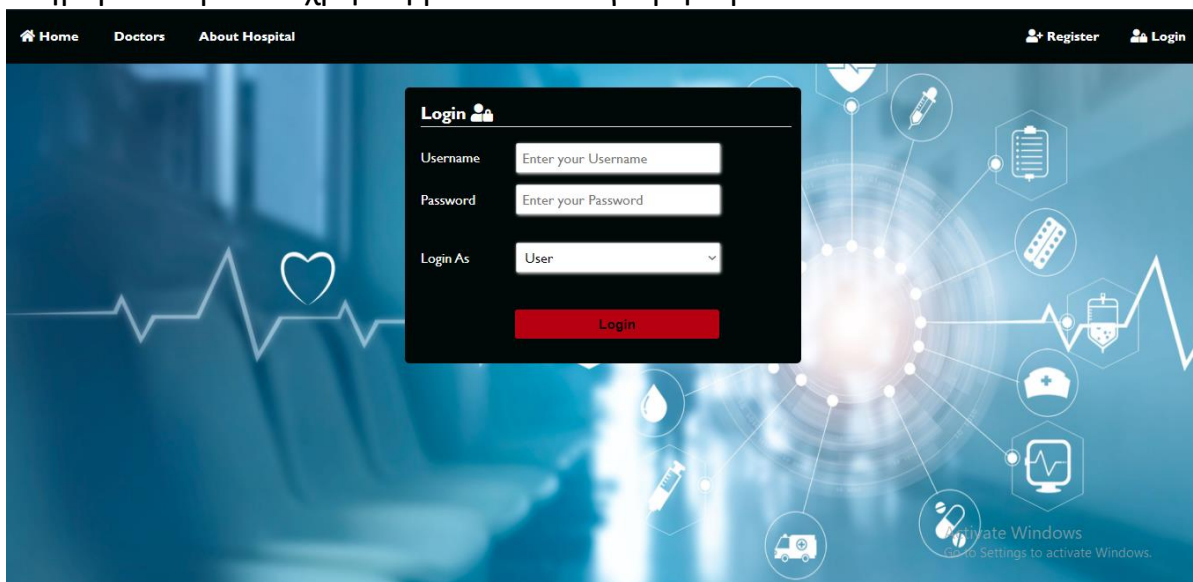
Ακολουθεί η αναλυτική περιγραφή του προγράμματος.

3.1 Κώδικας του αρχείου **Login.jsp** και **Login_Request.jsp**

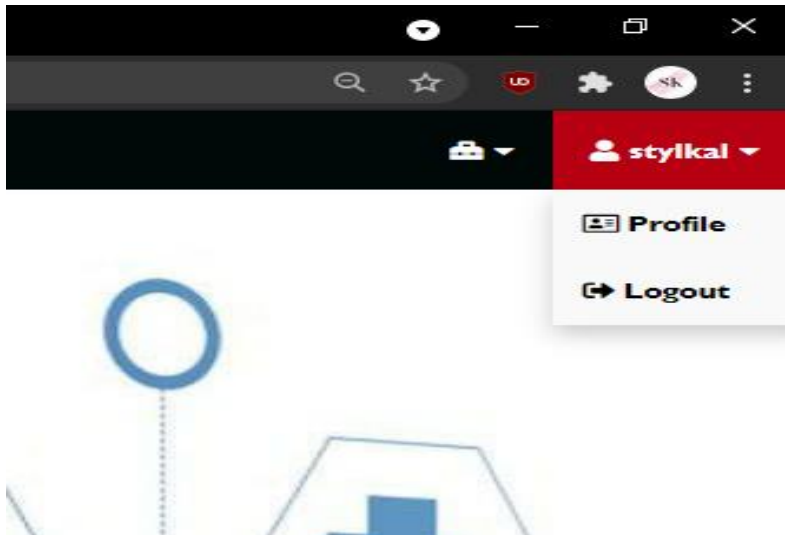
Στο αρχείο **Login.jsp** υπάρχει η function **<<formCheck()>>**, στην συνάρτηση γίνεται δήλωση μεταβλητών **username** και **password** αλλά και της λίστας



errors. Ακολουθώς γίνεται έλεγχος για το αν τα πεδία **username** και **password** που έχει συμπληρώσει ο χρήστης είναι κενά και γίνεται ανάλογη ενημέρωση στην λίστα **errors** αν υπάρξουν. Εφόσον μετά τον έλεγχο της, υπάρξει κάποιο λάθος ενημερώνουμε με το ανάλογο μήνυμα τον χρήστη. Το αρχείο **Login_Request.jsp** καλείται όταν γίνεται υποβολή της φόρμας στο αρχείο **login.jsp**. Παίρνουμε τις παράμετρος των πεδίων με την εντολή **request.getParameter**. Στην συνέχεια γίνεται σύνδεση στην βάση και προετοιμάζουμε ένα query το οποίο επιλέγει το **username** , **password** και **usertype** από την βάση το οποίο αντιστοιχίζει τις παράμετρος που έδωσε ο χρήστης. Αν ταυτίζονται μεταξύ τους τότε συνδέεται στην σελίδα αλλιώς ενημερώνουμε τον χρήστη με το ανάλογο μήνυμα.



Όταν κάνεις log in εμφανίζει τα ακόλουθα στα δεξιά:
-όνομα χρήστη: stylkal και για το παρόν στάδιο έχει λειτουργικότητα το Logout.



Αν κάνεις λάθος στην ταυτοποίηση εμφανίζει το ακόλουθο μήνυμα λάθους.



Εδώ βλέπουμε αν πατήσει στο Profile θα του εμφανίσει τα στοιχεία του κάθε χρήστη και μάλιστα μπορεί να κάνει και αλλαγές σε αυτά.



Rank: Administrator

First Name: styliani

Last Name: kalogirou

Email: stylkal@gmail.com

Telephone: 6947451700

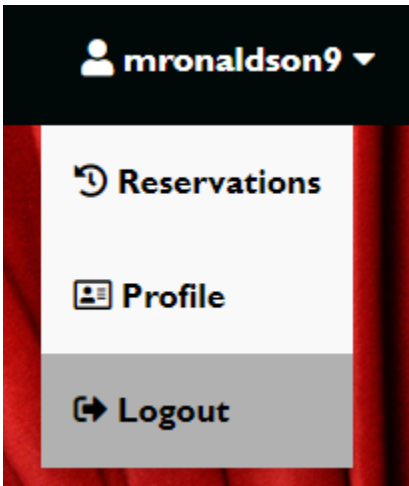
New Password: Leave empty if you don't wish to change

Current Password: Enter your current password

Save

3.1 Κώδικας του αρχείου `logout.jsp`

Όταν ο χρήστης επιλέξει την επιλογή του “Logout” στο menu τότε το session του θα καταστραφεί. Επίσης θα γίνει clear το cache με τις εντολές `setHeader` διότι το έγγραφο δεν πρέπει ποτέ να αποθηκεύεται προσωρινά και δεν πρέπει να αποθηκεύεται σε προσωρινή θέση στο δίσκο. Αυτή η κεφαλίδα προορίζεται για την αποφυγή ακούσιων αντιγράφων ευαίσθητων πληροφοριών.



3.2 Κώδικας του αρχείου Doctors.jsp

Συνδέουμε το αρχείο **doctors.jsp** με την βάση δεδομένων που έχουμε, με τον πίνακα **doctors**. Θα χρησιμοποιούμε τα queries για να εμφανίσουμε τα στοιχεία των γιατρών, τις ημερομηνίες και τις ώρες που θα έχουν ραντεβού. Για να επιτευχθούν τα παραπάνω, θα γίνεται η αντιστοίχιση των στοιχείων με βάση το **appointment_id** και ακολούθως εμφανίζονται ανά γραμμή τα ραντεβού με όλα τα στοιχεία.

3.3 Κώδικας των κλάσεων (Admins, Information, Members)

Αρχικά, όλες οι κλάσεις έχουν τις συναρτήσεις get οι οποίες επιστρέφουν String, Boolean, Integer και ArrayList για τα αποτελέσματα των queries. Για παράδειγμα εάν θέλω να βρω το ονοματεπώνυμο ενός χρήστη θα χρησιμοποιήσω την μέθοδο getName(username) που έχει σαν όρισμα το username του χρήστη και θα επιστρέφει String.

3.4 Κώδικας του αρχείου Panel_Users_View.jsp

Εάν ο χρήστης έχει την ιδιότητα ως διαχειριστής (Administrator), έχει την επιλογή με βάση το αρχείο **Panel_Users_View.jsp**, να εμφανίσει μια λίστα με όλους τους χρήστες, διαχειριστές περιεχομένου και διαχειριστές που έχουν λογαριασμό. Επίσης όχι μόνο μπορεί να δει τα στοιχεία τους αλλά και αν είναι συνδεδεμένη εκείνη την στιγμή.



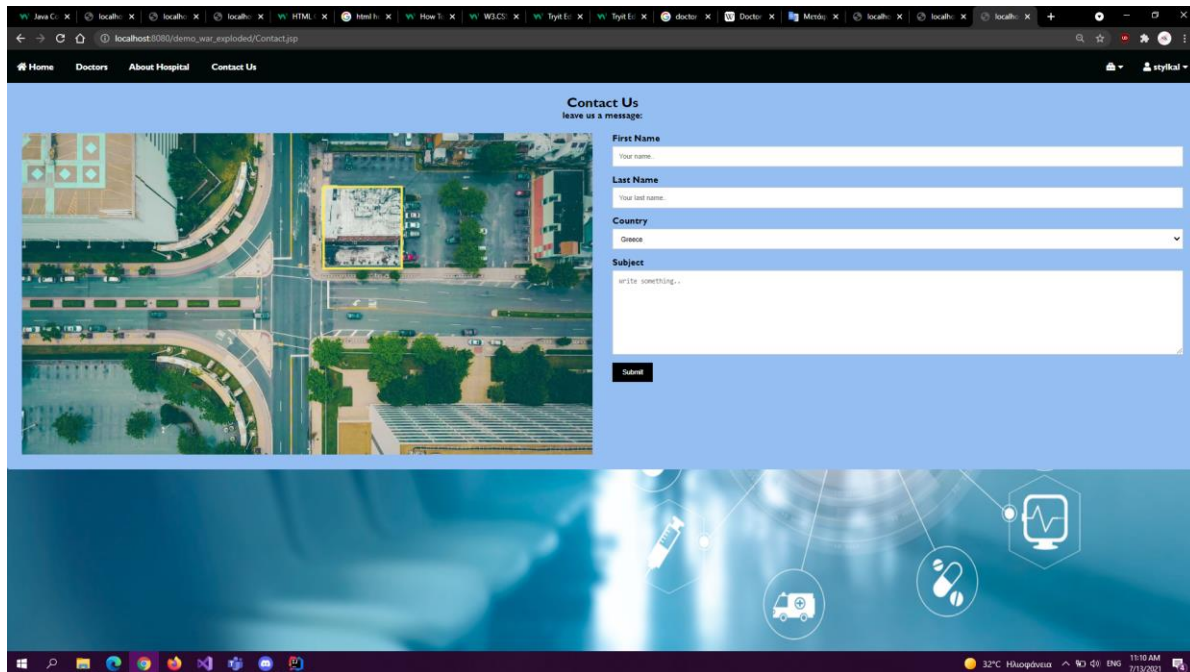
User ID:	Full Name:	Username:	Phone Number:	Usertype:	Email:
14	isavella	isavellapetrou	9678874584	User	isavel@hotmail.com
6	maria petrou	maria ptr	6985748521	User	maria petrou@gmail.com
7	stellap	stel	6956565632	User	stellapetrou@hotmail.com
9	xara	xarakal	6958396326	User	chara@gmail.com
3	petros con	petrosc	9876543465	User	petros@hotmail.com
10	georgekal	georgekalogirou	6958589656	User	georgekal@gmail.com
11	stavris	stavrosstavrou	6958958745	Doctor	stavros@gmail.com
13	maria x	maria xantou	6956265635	Doctor	maria@protonmail.com
1	stylani kalogirou	stylkal	6947451700	Administrator	stylkal@gmail.com
2	Charis	charis poly	99427173	Administrator	charis poly@gmail.com

3.5 Κώδικας του αρχείου Panel_Add_Appointment.jsp και

Αρχικά στο αρχείο Panel_Add_Appointment.jsp έχουμε μια φόρμα στην οποία ένας διαχειριστής μπορεί να προσθέσει μια ταινία στην βάση. Στην συνέχεια το αρχείο Panel_Appointment_Add_Request.jsp που θα καλεστεί όταν γίνει submit από την φόρμα του προηγούμενου αρχείου μέσω του action. Χρησιμοποιήσαμε ένα πεδίο τύπου hidden για να προσθέτουμε και να αφαιρούμε τις τιμές των checkboxes που είναι επιλεγμένες (checked). Μετά από αυτό παίρνουμε τις τιμές της προηγούμενης φόρμας και γίνεται έλεγχος εάν υπάρχει ήδη η ταινία μέσα στον πίνακα, επίσης, σε όλες τις σελίδες που ένας χρήστης πρέπει να είναι συνδεδεμένος για εκτελέσει τις λειτουργίες τους, γίνεται έλεγχος εάν ο χρήστης είναι συνδεδεμένος γιατί μπορεί να έληξε το session του ή να μην έχει τα κατάλληλα δικαιώματα (Δηλαδή να μην είναι admin, κτλ.). Επομένως θα εμφανιστεί το κατάλληλο μήνυμα και όλοι οι έλεγχοι είναι εντάξει θα προστεθεί η ταινία στον πίνακα.

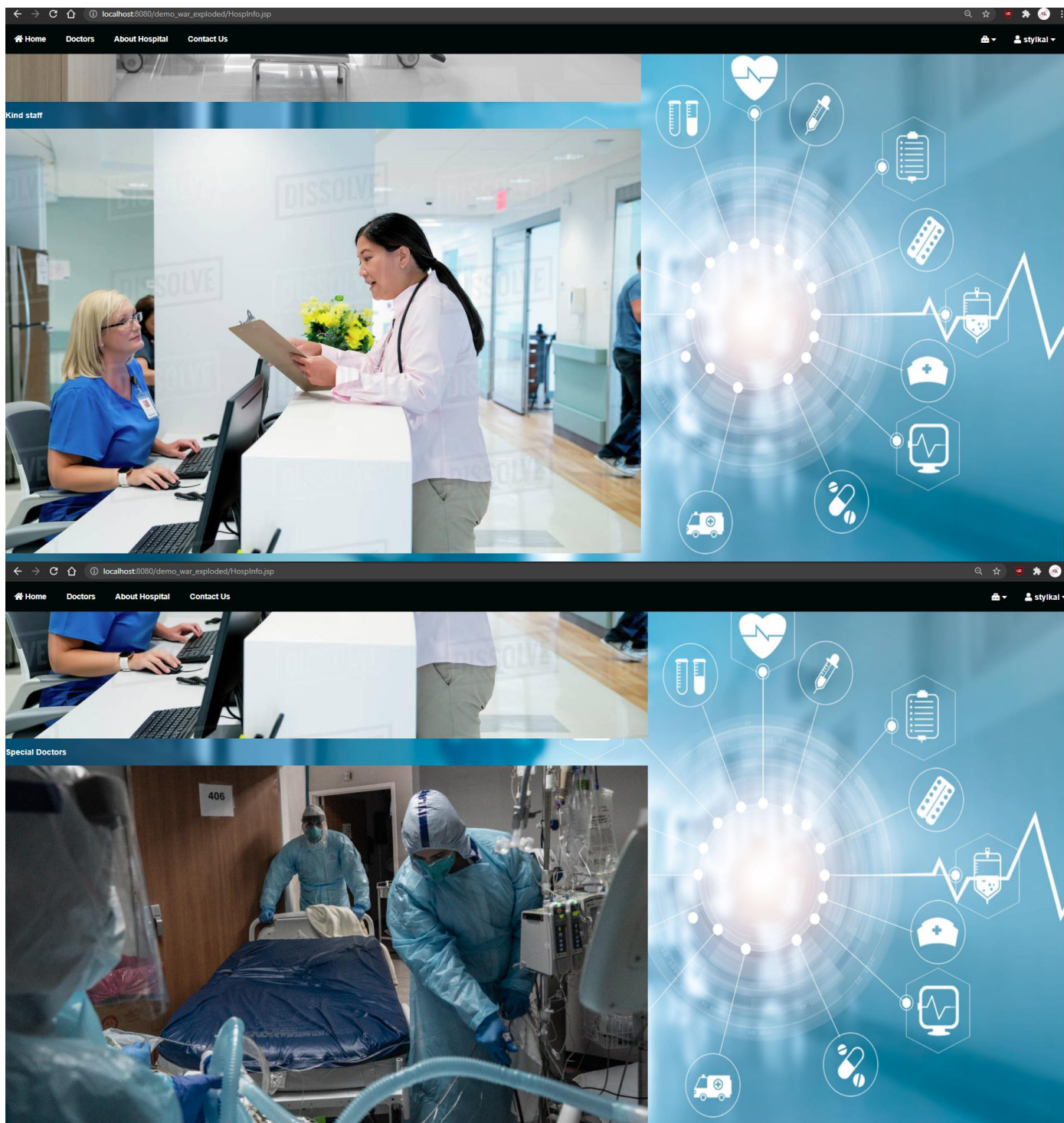


3.6 Κώδικας του αρχείου Contact.jsp



3.7 Κώδικας του αρχείου HospInfo.jsp







3.8 Κώδικας του αρχείου Doctors.jsp



4 Βιβλιογραφικές Πηγές

4.1 IntelliJ IDEA 2021.1

Είναι το IDE που χρησιμοποιήθηκε για την εργασία.

4.2 AdoptOpenJDK with Hotspot 8.212

Είναι το JDK που χρησιμοποιήθηκε για την εργασία
(<https://adoptopenjdk.net/>).



4.3 PGAdmin 4.8

4.4 PostgreSQL 10.5-1

4.5 Tomcat 9.0.35

<https://tomcat.apache.org/download-80.cgi>

4.6 PostgreSQL-42.2.5 (Driver)

<https://jdbc.postgresql.org/download.html>

4.7 SERVLETS και ΣΕΛΙΔΕΣ ΔΙΑΚΟΜΙΣΤΗ JAVA – ΤΕΧΝΟΛΟΓΙΕΣ ΠΥΡΗΝΑ 2^Η ΑΜΕΡΙΚΑΝΙΚΗ ΕΚΔΟΣΗ – MARTY HALL & LARRY BROWN

4.8 Συστήματα Βάσεων Δεδομένων 6^Η Έκδοση – Abraham Silberschatz, Henry F. Korth & S. Sudarshan

4.9 Menu

https://www.w3schools.com/css/css_navbar.asp

4.10 Ιδέες για τον τρόπο που θα γίνει το registration/login system

<https://www.javatpoint.com/registration-form-in-jsp>