**ΗΥ-359 Project**

* **Εισαγωγή σελίδα (2-3).**
* **Αναφορά στα servlets σελίδα (2-3).**
* **Αναφορά στα html σελίδα (3).**
* **Αναφορά στα JavaScript σελίδα (4-55).**
* **Αναφορά στον τρόπο συνδυασμού servlets-JavaScript-html σελίδα(6-10).**
* **Τέλος σελίδα (11).**

Στο πρότζεκτ δεν χρειάστηκε να αλλάξουμε την δοσμένη βάση, παρά μόνο να προσθέσουμε μερικές δικές μας συναρτήσεις στα EditTables που μας ήταν ήδη υλοποιημένα καθαρά για δικιά μας ευκολία.

Οι css που προσθέσαμε για κάθε καινούργιο html έχει αρκετά βασική υλοποίηση κυρίως για την όμορφη εμφάνιση της σελίδας.

Στο πρότζεκτ επίσης υλοποιήθηκαν και τα μπόνους τα οποία είναι:

1. Να στέλνει μήνυμα για όσους είναι για αιμοδοσία.
2. Αξιολόγηση γιατρών 1-5.

Στην παρακάτω σελίδα θα αναφερθούμε για το ποια **servlets** ορίσαμε τι τύπο τους δώσαμε (Get,Set,Post,Delete). Και ποιο ήταν περιληπτικά το σκεπτικό συνδυασμού τους με τον υπόλοιπο κώδικα. Αναλυτικότερα για τον συνδυασμό τους θα αναφερθούμε στο τέλος αφού εξηγήσουμε πρώτα όλα όσα πρέπει, περί servlets, JavaScript και html.

**~Servlets~**

* **Τα υλοποιημένα servlets είναι τα εξής:**

**Doctor:**

* 1. **AddAppointment(doPost):** Δημιουργεί ένα καινούργιο randevouz id με τα στοιχεία του γιατρού, null τα στοιχεία του ασθενή και status: free
  2. **CreateNewTreatment(doPost):** Δημιουργεί ένα καινούργιο treatment για τον ασθενή που μόλις εξέτασε και έθεσε το randevouz status: done
  3. **DocRandevous(doGet):** Μέσω του Doctor ID που είμαστε LoggedIn φορτώνουμε σε ένα json όλα τα randevouz και τα εμφανίζουμε μέσω JavaScript.
  4. **GetPatientID(doGet):** Μέσω του Doctor ID που είμαστε LoggedIn φορτώνουμε σε ένα json όλους τους Patients και τα εμφανίζουμε μέσω JavaScript.
  5. **GetRandevouzID(doGet):** Μέσω του Doctor ID που είμαστε LoggedIn φορτώνουμε σε ένα json όλους τους Patients και τα επεξεργαζόμαστε μέσω JavaScript.
  6. **ReturnDoctors(doGet):** Επιστρέφουμε όλους τους doctors σε μορφή json και το επεξεργαζόμαστε μέσω JavaScript.

**Patient:**

1. **ActiveTreatments(doGet):** Μέσω του Patient ID που είμαστε LoggedIn φορτώνουμε σε ένα json όλα τα treatments του χρήστη μας και τα εμφανίζουμε μέσω JavaScript.
2. **InsertNewBloodTest(doPost):** Σε περίπτωση που ο χρήστης μας θέλει να δηλώσει καινούργιο blood test δηλώνει τα σημαντικά στοιχεία στο form και δημιουργούμαι μια καινούργια τιμή στο bloodtest table.
3. **UserRandevouz(doGet):** Μέσω του Patient ID που είμαστε LoggedIn φορτώνουμε σε ένα json όλα τα randevouz του χρήστη μας και τα επεξεργαζόμαστε μέσω JavaScript.
4. **patientBTinfo(doGet):** Μέσω του Patient ID που είμαστε LoggedIn φορτώνουμε σε ένα json όλα τα blood tests του χρήστη μας και τα επεξεργαζόμαστε μέσω JavaScript.

**Both:**

1. **SignIn(doPost):** Συμπληρώνοντας την SignIn φόρμα ο χρήστης δημιουργεί μια νέα τιμή είτε στο users ή στο doctors table.
2. **LogIn(Both):** Βάζοντας τα σωστά credentials(username, password) που του χρήστη εισέρχεται στο προφίλ του.
3. **LogOut(doPost):** Εφόσον κάνει logout λήγει η συνεδρία και θα πρέπει να ξανακάνει login για να ξαναμπεί στο προφίλ του.
4. **ChangeProfileData(doPost):** Αλλάζει τα στοιχεία του χρήστη στα οποία υπάρχουν αλλαγές.
5. **UpdateRandevouState(doPost):** Ανάλογα με την ενέργεια που έχει προκύψει αλλάζει και το state του randevou στην κατάσταση που του αρμόζει (free, selected, done, canceled).

Στην παρακάτω σελίδα θα αναφερθούμε πάνω στην html, πως δομήθηκε και πως την συνδυάσαμε με τον υπόλοιπο κώδικα.

**~HTML ~**

* **Html κώδικας που προστέθηκε:**

**Patient:**

1. **userAppSelect:** Μέσω του Patient ID που είμαστε LoggedIn φορτώνουμε σε ένα json όλα τα treatments του χρήστη μας και τα εμφανίζουμε μέσω JavaScript.
2. **userBloodTest:** έχει όλες τις πληροφορίες όλων των bloodtest που έχει κάνει ο συγκεκριμένος χρήστης.
3. **userTreatments:** Δείχνει όλα τα treatments που έχουν πιστοποιηθεί από γιατρούς για τον συγκεκριμένο χρήστη.

**Both:**

1. **userpage/** **docpage:** html page μετά από το επιτυχές login του επισκέπτη.
2. **userAppontments/docAppointments:** Εμπεριέχει τα appointments του συγκεκριμένου χρήστη με τον/τους αντίστοιχο/ους εξεταζόμενους/εξεταστές του.

Αργότερα εξηγούμε πως αυτά περνάνε πληροφορίες στην αντίστοιχη συνάρτηση στην JavaScript και πως εμείς την αξιοποιούμαι.

Στην επόμενη σελίδα θα μιλήσουμε για την JavaScript, πως δομήθηκε ποιες είναι οι συναρτήσεις τις. Μετά από αυτό θα αναφερθούμε ποια είναι η ιδέα πίσω από κάθε συνάρτηση και πως αυτή συνεργάζεται με το υπόλοιπο πρόγραμμα.

**JavaScript**

* **Προστέθηκαν οι συναρτήσεις:**
* **-- connection to servlet Post**
  1. AddAppointment(){}
  2. CreateNewTreatment(){}
  3. InsertNewBloodTest(){}
  4. UpdateRandevouToDone(){}
  5. BookAppointment(){}
* **-- connection to servlet Get**
  1. GetPatientID(){}
  2. GetRandevouzID(){}
  3. createDocViewAppointments(){}
  4. patientBTinfo(){}
  5. showless(showless is a function that calls updateRandevouToCancelled()) && UpdateRandevouToCanceled(){}
  6. User\_ActiveTreatments(){}
  7. AllDocRandevouz(){}
  8. ViewAppoint(ViewAppoint is a function that calls AllUserRandevouz ()) && AllUserRandevouz(){}
  9. User\_Treatments(){}
* **-- Create dynamic Table**
  1. goBloodTest(){}
  2. getPatientTreatments(){}
  3. ViewBT(){}
  4. ShowPatientBT2(){}
  5. DoctorsTable2(){}
  6. createTableFromJSON2(){}
* **-- API’s and other**
  1. PDFfunction(){}
  2. API\_doctors\_dest(){}
  3. rankingSorter(){}
  4. sort\_doc\_by\_val(){}
  5. createDocMap(){}
  6. drawChart(){}

**Πως γίνονται οι διάφορες αυτές ενώσεις αναλυτικά;**

* **🡨** αυτό το σύμβολο από εδώ και πέρα θα αναφέρεται στην JavaScript συνάρτηση
  + **🡨** ενώαυτό το σύμβολο από εδώ και πέρα θα αναφέρεται στα δεδομένα html/servlet που χρησιμοποιεί αυτή η συνάρτηση.
    - **POST**
* AddAppointment():
  + **docAppointments-html**- Αφού ο χρήστης μας συμπληρώσει την φόρμα το html με το submit θα στείλει την φόρμα στο JavaScript μας και αυτό με την σειρά του θα καλέσει το servlet AddAppointment
  + **AddAppointment()-servlet**- Αφού καλεστεί το method Post του servlet θα πάρουμε όλα τα στοιχεία της φόρμας και θα δημιουργήσουμε ένα καινούργιο ραντεβού με status free στην βάση εφόσον η ημερομηνία του είναι μεταγενέστερη της δικής μας.
* CreateNewTreatment()
  + **createDocViewAppointments()-JavaScript-** είναι μια δυναμική φόρμα όπου εμφανίζει όλους τους ασθενείς status: selected του συγκεκριμένου γιατρού.
  + **CreateNewTreatment()-servlet-**Παίρνει τα στοιχεία που του δίνονται από το createDocViewAppointments και το αντίστοιχο doctorID και userID(ανάλογα με ποιον επιλέξαμε) από το xhr.send. Αναλαμβάνει με την σειρά του να δημιουργήσει μέσω της μεθόδου POST να δημιουργήσει ένα καινούργιο treatment για τον ασθενή με το αντίστοιχο userID και bloodestID.
* InsertNewBloodTest()
  + **userBloodTest-html**- Αφού ο χρήστης μας συμπληρώσει την φόρμα το html με το submit θα στείλει την φόρμα στο JavaScript μας και αυτό με την σειρά του θα καλέσει το servlet InsertNewBloodTest
  + **InsertNewBloodTest()-servlet-** Αφού καλεστεί το method Post του servlet θα πάρουμε όλα τα στοιχεία της φόρμας και θα δημιουργήσουμε ένα καινούργιο bloodtest στην βάση.
* UpdateRandevouToDone() && BookAppointment() && UpdateRandevouToCanceled()
  + **UpdateRandevouState()-servlet-**Καλείται και με τις σωστές παραμέτρους ενημερώνει το state του αντίστοιχου randevou σε selected,free,done.
    - **GET**
* GetPatientID()
  + **GetPatientID()-servlet-** Μέσω του doctorID επιστρέφει μέσα σε ένα array Json όλους τους ασθενής και τις πληροφορίες τους που έχει αναλάβει ο συγκεκριμένος γιατρός.
* GetRandevouzID()
  + **GetRandevouzID-servlet-** Μέσω του doctorID επιστρέφει μέσα σε ένα array Json όλα τα randevouz που έχει αναλάβει ο συγκεκριμένος γιατρός τις πληροφορίες του χρήστη και τις πληροφορίες του ασθενή.
* createDocViewAppointments()
  + Δημιουργεί ένα δυναμικό table οπού ο γιατρός θα μπορεί να βλέπει όλα του τα ραντεβού με τους ασθενής.
* patientBTinfo()
  + **patientBTinfo()-servlet-** Επιλέγει όλα τα bloodtest και τα επιστρέφει σε μορφή json.
* User\_ActiveTreatments()
  + Περιέχει ένα δυναμικό table από τα treatments του χρήστη (εάν υπάρχουν) και καλεί το servlet ActiveTreatments
  + **ActiveTreatments()-servlet -** Επιστρέφει όλα τα treatments όπου το id είναι userID.
* AllDocRandevouz()
  + Περιέχει ένα δυναμικό table με όλα τα διαθέσιμα ραντεβού του συγκεκριμένου γιατρού.
  + **DocRandevous()-servlet -** Επιστρέφει όλα τα randevouz όπου το id είναι doctorID.
* AllUserRandevouz()
  + Περιέχει ένα δυναμικό table από τα ραντεβού του ασθενή και τα εμφανίζει διαφορετικά ανάλογα με την κατάσταση του ραντεβού.
  + **UserRandevouz()-servlet -** Παίρνει όλα τα ραντεβού όπου userID = id.
* User\_Treatments()
  + **ActiveTreatments()-servlet -** Επιστρέφει όλα τα treatments όπου το id είναι userID. Αλλά αυτήν την φορά για τον user.
    - **Dynamic Tables**
* goBloodTest(){}
  + αλλάζει html και ενημερώνει το όνομα και ΑΜΚΑ του ασθενή στην σελίδα.
* getPatientTreatments(){}
  + Ένα δυναμικό table όπου εμφανίζει τις απαραίτητες πληροφορίες για κάθε treatment του χρήστη.
* ViewBT(){}
  + Ένα δυναμικό table όπου εμφανίζει τις απαραίτητες πληροφορίες για κάθε bloodtest του χρήστη από το login του doctor.
* ShowPatientBT2(){}
  + Ένα δυναμικό table όπου εμφανίζει τις απαραίτητες πληροφορίες για κάθε bloodtest του χρήστη από το login του ασθενή.
* DoctorsTable2(){}
  + Ένα δυναμικό table όπου εμφανίζει όλους τους doctors διαβάζοντας το json που λαμβάνει από το ReturnDoctors.
  + **ReturnDoctors()-servlet**
* createTableFromJSON2(){}
  + Εμφανίζει όλους τους γιατρούς που είναι certified μέσω ενός δυναμικού table.
    - **API’s and other**
* PDFfunction()
  + Κάναμε insert την βιβλιοθήκη:

<” https://cdnjs.cloudflare.com/ajax/libs/jspdf/1.5.3/jspdf.min.js”> και μετά το συνδυάσαμε το JavaScript έτσι ώστε όταν το κάνουμε κλικ να κατεβάζουμε το pdf της συγκεκριμένης μέρας.

* API\_doctors\_dest()
* rankingSorter()
* sort\_doc\_by\_val()
* createDocMap()
* drawChart()

Δεν αναφέρουμε servlets html’s και JavaScripts functions που είχαν οριστεί από τις προηγούμενες σειρές ασκήσεων για απλοποίηση της αναφοράς. Επίσης δεν υπήρξε υλοποίηση του REST δεδομένου ότι το υλοποιήσαμε με servlets.

**Βιβλιοθήκες που χρησιμοποιήθηκαν:**

1. <https://cdnjs.cloudflare.com/ajax/libs/jspdf/1.5.3/jspdf.min.js> -- **pdf**
2. <https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js> -- **queries**
3. <http://www.openlayers.org/api/OpenLayers.js> -- **Map**
4. <https://www.gstatic.com/charts/loader.js> --

**Ασχολήθηκαν οι:**

1. **Ιωάννης Θεοχαράκης CSD 4263**
2. **Κωνσταντίνος Τσανακτσής CSD 4157**