



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Πρόγραμμα Μεταπτυχιακών Σπουδών**

**«Πληροφορικής»**

**Μεταπτυχιακή Διατριβή**

Τίτλος Διατριβής	<b>Εφαρμογή «Ηλεκτρονικοί φάκελοι ασθενών» Application “Electronic Medical Records”</b>
Όνοματεπώνυμο Φοιτητή	<b>ΤΣΙΧΛΗΣ ΧΡΗΣΤΟΣ ΝΕΚΤΑΡΙΟΣ</b>
Πατρώνυμο	<b>ΠΑΝΤΕΛΗΣ</b>
Αριθμός Μητρώου	<b>ΜΠΠΛ20084</b>
Επιβλέπων	<b>Ευθύμιος Αλέπης, Αναπληρωτής Καθηγητής</b>

Δεκέμβριος 2022

---

**Τριμελής Εξεταστική Επιτροπή**

Ευθύμιος Αλέπης  
Αναπληρωτής Καθηγητής

Μαρία Βίβου  
Καθηγήτρια

Ευάγγελος Σακκόπουλος  
Αναπληρωτής Καθηγητής

## Περιεχόμενα

<b>ΠΕΡΙΛΗΨΗ .....</b>	<b>3</b>
<b>ABSTRACT .....</b>	<b>3</b>
<b>1. ΕΙΣΑΓΩΓΗ.....</b>	<b>4</b>
<b>2. ΑΝΑΣΚΟΠΗΣΗ ΠΕΔΙΟΥ .....</b>	<b>4</b>
<b>3.ΠΑΡΟΥΣΙΑΣΗ ΚΑΙ ΧΡΗΣΗ ΕΦΑΡΜΟΓΗΣ .....</b>	<b>5</b>
<b>3.1. Desktop application .....</b>	<b>5</b>
3.1.1. Patients .....	7
3.1.2. Patient History .....	9
3.1.3. Blood Section .....	10
3.1.4. Prescriptions .....	12
3.1.5. Appointments .....	16
<b>3.2. Web application.....</b>	<b>17</b>
<b>3.2.1. Ιατρός.....</b>	<b>19</b>
Patients .....	20
History of patients! .....	22
Blood information! .....	23
<b>3.2.2. Ασθενείς.....</b>	<b>25</b>
<b>4.ΑΝΑΛΥΣΗ ΤΟΥ ΚΩΔΙΚΑ.....</b>	<b>26</b>
<b>4.1. Desktop application .....</b>	<b>26</b>
4.1.1. Form1.....	27
4.1.2. Menu.....	28
4.1.3 Patient Form .....	29
4.1.4. Patient History .....	35
4.1.5. Prescriptions .....	41
4.1.6. Appointments .....	45

<b>4.2.Web application.....</b>	<b>46</b>
4.2.1.Login.....	46
4.2.2.Sign up .....	47
4.2.3.Menu.....	49
4.2.4.Patients .....	50
4.2.5.History.....	52
4.2.6.Blood stats .....	54
4.2.7.Appointments .....	58
<b>5.ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....</b>	<b>60</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>61</b>

## ΠΕΡΙΛΗΨΗ

Η συγκεκριμένη εφαρμογή βοηθά ιατρούς με δικά τους ιδιωτικά ιατρεία να καταχωρούν τα προσωπικά στοιχεία των ασθενών τους ενώ μπορούν επίσης να καταχωρούν πληροφορίες για τις αιματολογικές εξετάσεις που τους προσκομίζουν οι ίδιοι οι ασθενείς. Επιπλέον, δίνει τη δυνατότητα στους ιατρούς να παρέχουν συνταγογραφίες, να τυπώνουν τις αιματολογικές εξετάσεις των ασθενών, καθώς και να ελέγχουν τα ραντεβού που έχουν για κάθε μέρα. Η εφαρμογή περιλαμβάνει και webκομμάτι, στο οποίο μπορούν οι ασθενείς του ιατρού να κάνουν εγγραφή με τα στοιχεία τους δημιουργώντας έναν λογαριασμό, με τον οποίο μπορούν να κλείσουν ηλεκτρονικά ραντεβού ή ακόμα και να το ακυρώσουν. Επιπλέον στο διαδικτυακό κομμάτι, ο ιατρός μπορεί ξανά να καταχωρεί τα παραπάνω δεδομένα για τους ασθενείς του, με τη διαφορά ότι δεν μπορεί να έχει πρόσβαση στις συνταγογραφίες και στα ραντεβού που εκκρεμούν.

## ABSTRACT

This application helps doctors with their own private practices to register the personal details of their patients while they can also register information about the blood tests provided by the patients themselves. In addition, it enables doctors to provide prescriptions, print patients' blood tests, as well as check their appointments for each day. The application also includes a web part, in which the doctor's patients can register with their details by creating an account, with which they can make an online appointment or even cancel it. In addition, in the online part, the doctor can again enter the above data for his patients, with the difference that he cannot access prescriptions and pending appointments.

## 1. ΕΙΣΑΓΩΓΗ

Σκοπός της παρούσας πτυχιακής εργασίας είναι η δημιουργία και σωστή λειτουργία 2 παρόμοιων εφαρμογών web-desktop applications για τη διαχείριση και καταχώρηση ιατρικών στοιχείων ασθενών από τους ιατρούς τους, καθώς και η δυνατότητα προσκόμισης συνταγογραφίσεων από τη μεριά των γιατρών και η δυνατότητα από τη μεριά των ασθενών να κλείνουν ηλεκτρονικά ραντεβού.

Η εργασία αυτή απευθύνεται κυρίως σε ιατρούς με ιδιωτικά ιατρεία και δεν προορίζεται για μεγαλύτερα ιατρικά κέντρα (π.χ νοσοκομεία) καθώς το πλήθος πληροφοριών που χειρίζεται δεν είναι αρκετό για αυτά. Επιπλέον μπορεί να χρησιμοποιηθεί από κάθε πολίτη που επιθυμεί να κλείνει ραντεβού ηλεκτρονικά με τα ιατρεία που θα χρησιμοποιούν αυτή την εφαρμογή.

Για την εκπόνηση της συγκεκριμένης εργασίας χρησιμοποιήθηκαν οι εξής γλώσσες προγραμματισμού και βάσεις δεδομένων:

- C# visual programming language, Χρησιμοποιώντας το πρόγραμμα Microsoft Visual Studio 2019 (Windows Forms App (.NET Framework), ASP.NET Web Application (.NET Framework))
- HTML 5 – Bootstrap
- CSS
- JavaScript
- Microsoft SQL Server Management Studio 18

## 2. ΑΝΑΣΚΟΠΗΣΗ ΠΕΔΙΟΥ

Στην αγορά υπάρχουν ήδη προγράμματα που χρησιμοποιούνται από νοσοκομεία και ιατρεία για την αποθήκευση πληροφοριών ασθενών και των ιατρικών τους φακέλων. Επιπλέον πολλά από αυτά τα προγράμματα συνδέονται και με ένα σύστημα τιμολόγησης για τις υπηρεσίες που παρέχουν, προσθέτοντας ακόμα περισσότερες πληροφορίες που δεν σχετίζονται απαραίτητα με την υγεία των ασθενών. Η εφαρμογή που δημιουργήθηκε για τη μεταπτυχιακή διατριβή μπορεί να μην παρέχει το ίδιο μεγάλο πλήθος πληροφοριών που προσφέρουν άλλες ήδη υπάρχουσες εφαρμογές, αλλά οι πληροφορίες και λειτουργίες που προσφέρει είναι εύκολα προσβάσιμες, σημαντικές, καθώς και είναι εύκολη στη χρήση και από τη μεριά των γιατρών αλλά και των ασθενών-χρηστών. Παραδείγματα εφαρμογών που υπάρχουν ήδη στην αγορά από τις οποίες εμπνεύστηκε η συγκεκριμένη εφαρμογή είναι:

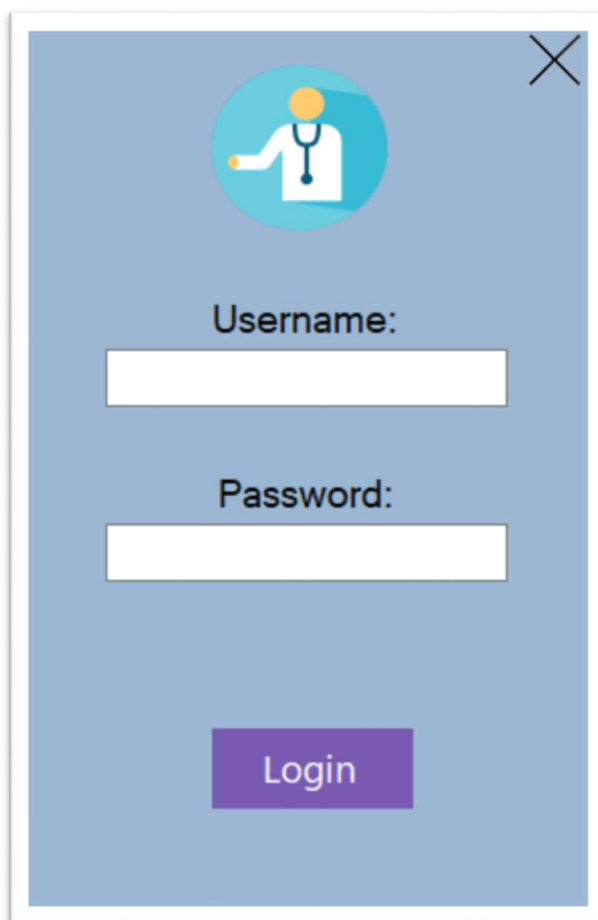
- athenaOne (athenahealth)
- Eclipse Practice Management Software

- MyHealth app (gov.gr)
- Epic software (Epic systems Corporation)
- MEDITECH

### 3.ΠΑΡΟΥΣΙΑΣΗ ΚΑΙ ΧΡΗΣΗ ΕΦΑΡΜΟΓΗΣ

#### 3.1. Desktop application

Ανοίγοντας την εφαρμογή μας παρουσιάζεται το παρακάτω παράθυρο (βλ. Εικόνα 1):



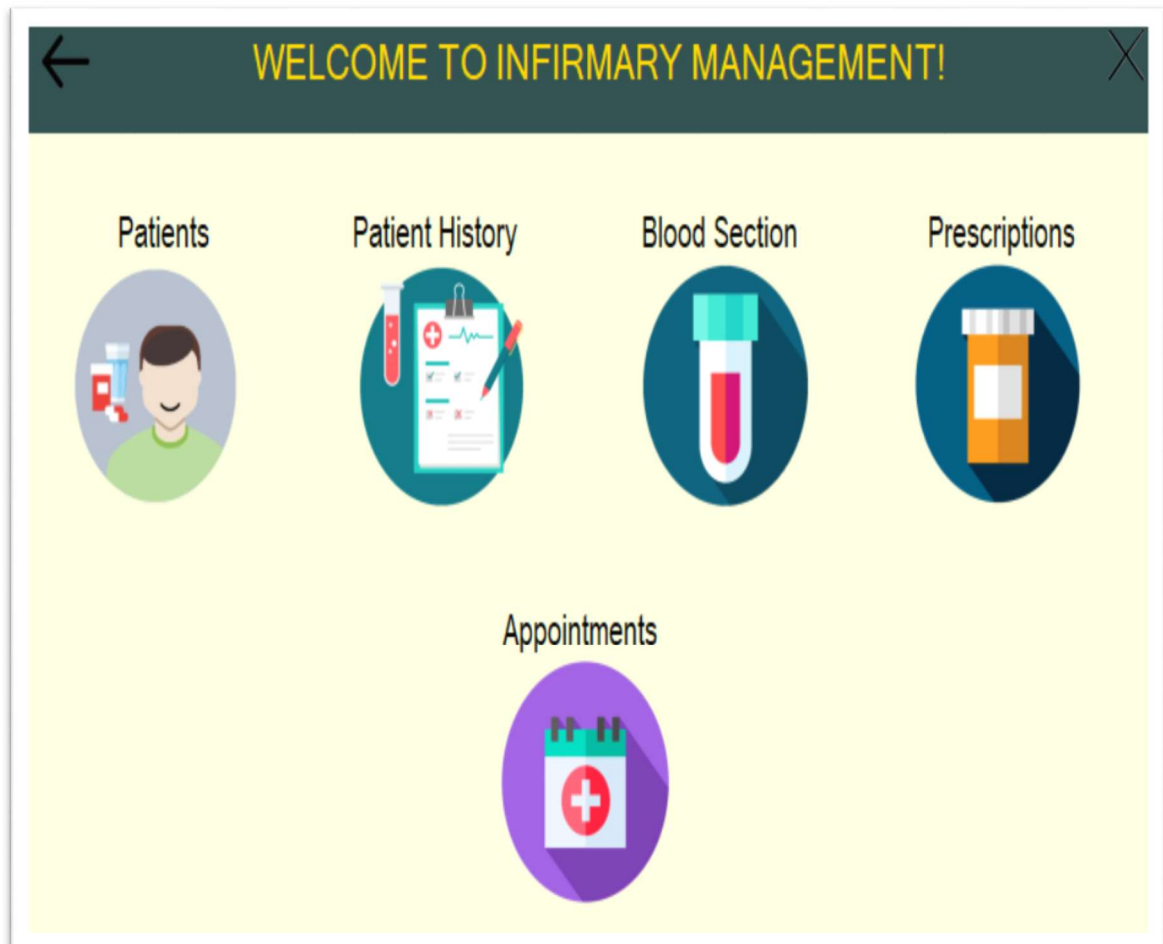
Εικόνα 1 Login Screen

Στο παράθυρο αυτό ο ιατρός καλείται να βάλει τα στοιχεία του λογαριασμού του username-password για να αποκτήσει πρόσβαση στις επιπλέον λειτουργίες του προγράμματος. Τα στοιχεία του λογαριασμού του ιατρού είναι:

Username: admin

Password: 135

Πατώντας το κουμπί **Login**, αν ο ιατρός έχει δώσει τα σωστά στοιχεία θα μεταφερθεί στο βασικό μενού της εφαρμογής (βλ. Εικόνα 2):



**Εικόνα 2 Menu**

Υπάρχουν 5 κατηγορίες τις οποίες μπορεί να επιλέξει ο ιατρός τις οποίες θα αναλύσουμε στη συνέχεια:

1. Patients
2. Patient History
3. Blood Section
4. Prescriptions
5. Appointments



Επιπλέον σε κάθε φόρμα της εφαρμογής πατώντας το εικονίδιο X στη πάνω δεξιά μεριά κάθε φόρμας τερματίζουμε τη λειτουργία της εφαρμογής.

Παρακάτω περιγράφονται οι λειτουργίες κάθε κατηγορίας του μενού:

### 3.1.1. Patients

Πατώντας στο εικονίδιο της συγκεκριμένης κατηγορίας ο ιατρός μεταφέρεται στη παρακάτω φόρμα (βλ. Εικόνα 3):

The screenshot shows a mobile application form titled "ADD NEW OR UPDATE PATIENT INFO!". The form is set against a light blue background with a dark green header. The header contains a back arrow on the left and a close 'X' icon on the right. The form fields are arranged as follows: "Full Name" (text input), "Age" (text input) and "Gender" (dropdown menu) are in the top row. "Phone Number" (text input) is in the second row. "Height" (text input) and "Weight" (text input) are in the third row. "Blood Type" (dropdown menu) is in the fourth row. "AMKA (must be unique)" (text input) and "Address" (text input) are in the fifth and sixth rows. On the right side, there are three text input fields labeled "Symptoms", "Diagnosis", and "Medicines". In the center of the form, there are three buttons: "Save", "Update", and "Delete".

Εικόνα 3 Patients form

Στη συνέχεια ο ιατρός καλείται να συμπληρώσει έναν συγκεκριμένο αριθμό από τα διαθέσιμα textboxes που δίνονται ανάλογα με το σκοπό του.

Αν επιθυμεί να αποθηκεύσει ένα νέο ασθενή στο σύστημα τότε θα πρέπει να συμπληρώσει όλα τα κουτιά και στη συνέχεια να πατήσει το κουμπί **'Save'**.

Αν επιθυμεί να ενημερώσει τις πληροφορίες κάποιου ασθενή που υπάρχει ήδη στο σύστημα τότε πρέπει υποχρεωτικά να έχει συμπληρώσει το κουτί ΑΜΚΑ με το σωστό αριθμό του ασθενή που υπάρχει στη βάση, και στη συνέχεια να συμπληρώσει από 1 έως όλα τα υπόλοιπα κουτιά των οποίων τις πληροφορίες επιθυμεί να ενημερώσει. Αφού υλοποιήσει σωστά τις παραπάνω διαδικασίες, πατάει το κουμπί **'Update'** για να τις αποθηκεύσει στο σύστημα.

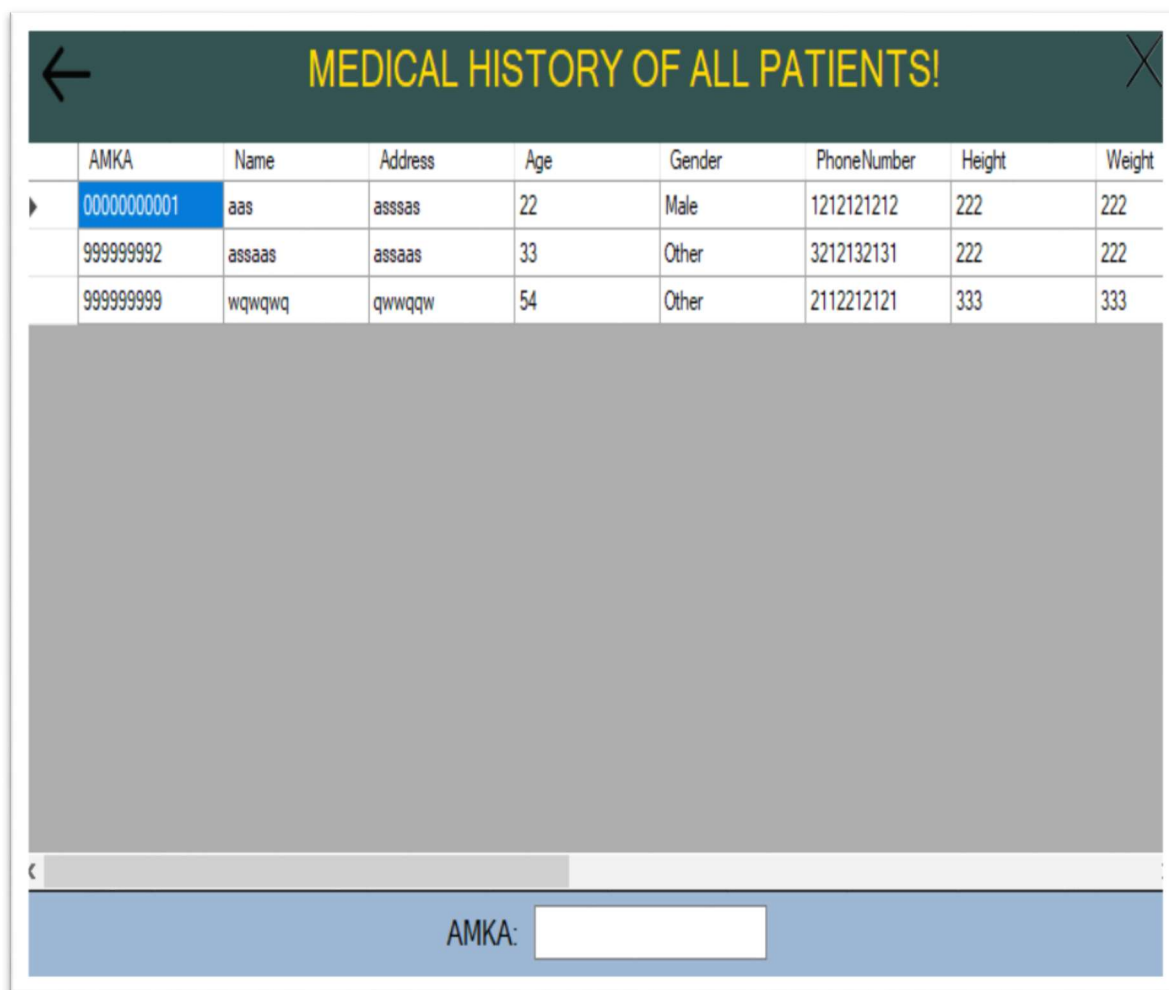
Αν τέλος επιθυμεί να διαγράψει έναν ήδη υπάρχον ασθενή από το σύστημα, τότε το μόνο που έχει να κάνει είναι να πληκτρολογήσει στο κουτί ΑΜΚΑ τον αριθμό φορολογικού μητρώου του ασθενή του οποίου θέλει να διαγράψει από το σύστημα και στη συνέχεια να πατήσει το κουμπί **'Delete'**. **ΠΡΟΣΟΧΗ!!!** Εδώ θα πρέπει να σημειωθεί ότι αυτή η διαδικασία δεν αναιρείται και διαγράφει πλήρως οποιαδήποτε πληροφορία σχετίζεται με τον ΑΜΚΑ του συγκεκριμένου ασθενή από κάθε πίνακα της βάσης δεδομένων μας.

Παρακάτω αναλύουμε τι πληροφορίες πρέπει να περιέχει κάθε textbox:

- Full Name**: ονοματεπώνυμο
- Age**: ηλικία
- Gender**: φύλο
- Phone Number**: αριθμός τηλεφώνου
- Height**: ύψος
- Weight**: βάρος
- Blood Type**: τύπος αίματος
- ΑΜΚΑ**: αριθμός μητρώου κοινωνικής ασφάλισης
- Address**: διεύθυνση κατοικίας
- Symptoms**: συμπτώματα
- Diagnosis**: διάγνωση του ιατρού
- Medicines**: φάρμακα που προτείνει ο ιατρός

### 3.1.2. Patient History

Πατώντας το εικονίδιο της συγκεκριμένης κατηγορίας ο ιατρός μεταφέρεται στη παρακάτω φόρμα (βλ. Εικόνα 4):



Εικόνα 4 Patient History form

Εδώ ο ιατρός έχει τη δυνατότητα να δει τις βασικές πληροφορίες για κάθε ασθενή που έχει καταχωρήσει στο σύστημα, και χρησιμοποιώντας το τεκτβοχστη κάτω πλευρά της φόρμας μπορεί να κάνει αναζήτηση ενός συγκεκριμένου AMKA για να δει αμέσως τις πληροφορίες για τον ασθενή με το AMKA που πληκτρολόγησε στο κουτί.

### 3.1.3. Blood Section

Πατώντας το εικονίδιο της συγκεκριμένης κατηγορίας ο ιατρός μεταφέρεται στη παρακάτω φόρμα (βλ. Εικόνα 5):

**BLOOD INFO ABOUT PATIENTS!**

AMKA:

**White Cells**

WBC:  NEUT(%):

LYMPH(%):  MONO(%):

EOS(%):  BASO(%):

**Red Cells**

RBC:  HGB:  RDW:

HCT:  MCV:

MCH:  MCHC:

**Platelets**

PLT:  PCT:

MPV:  PDW:

Save Update Delete

AMKA	WBC	NEUT
▶ 00000000001	5	5
999999992	1	2
999999999	1	1.1

WBC	NEUT	LYMPH
▶ 4.00 - 10.50	40.0 - 75.0	20.0 - 45.0

AMKA:

Εικόνα 5 Blood Info form

Ο κάτω αριστερά πίνακας δείχνει όλους τους ασθενείς για τους οποίους ο ιατρός έχει καταχωρήσει εξετάσεις αίματος στο σύστημα, και παρόμοια με το κουτί που βρίσκεται κάτω από τον συγκεκριμένο πίνακα ο ιατρός πληκτρολογώντας έναν AMKA ενός ασθενή που

υπάρχει στη βάση μπορεί να δει τις πληροφορίες για το αίμα του. Ο πίνακας κάτω δεξιά δείχνει τις τιμές τις οποίες πρέπει να έχει κάθε στοιχείο του αίματος για να θεωρείται 'εντός ορίων'. Ο ιατρός επίσης καλείται να συμπληρώσει έναν συγκεκριμένο αριθμό από τα διαθέσιμα text boxes που δίνονται ανάλογα με το σκοπό του.

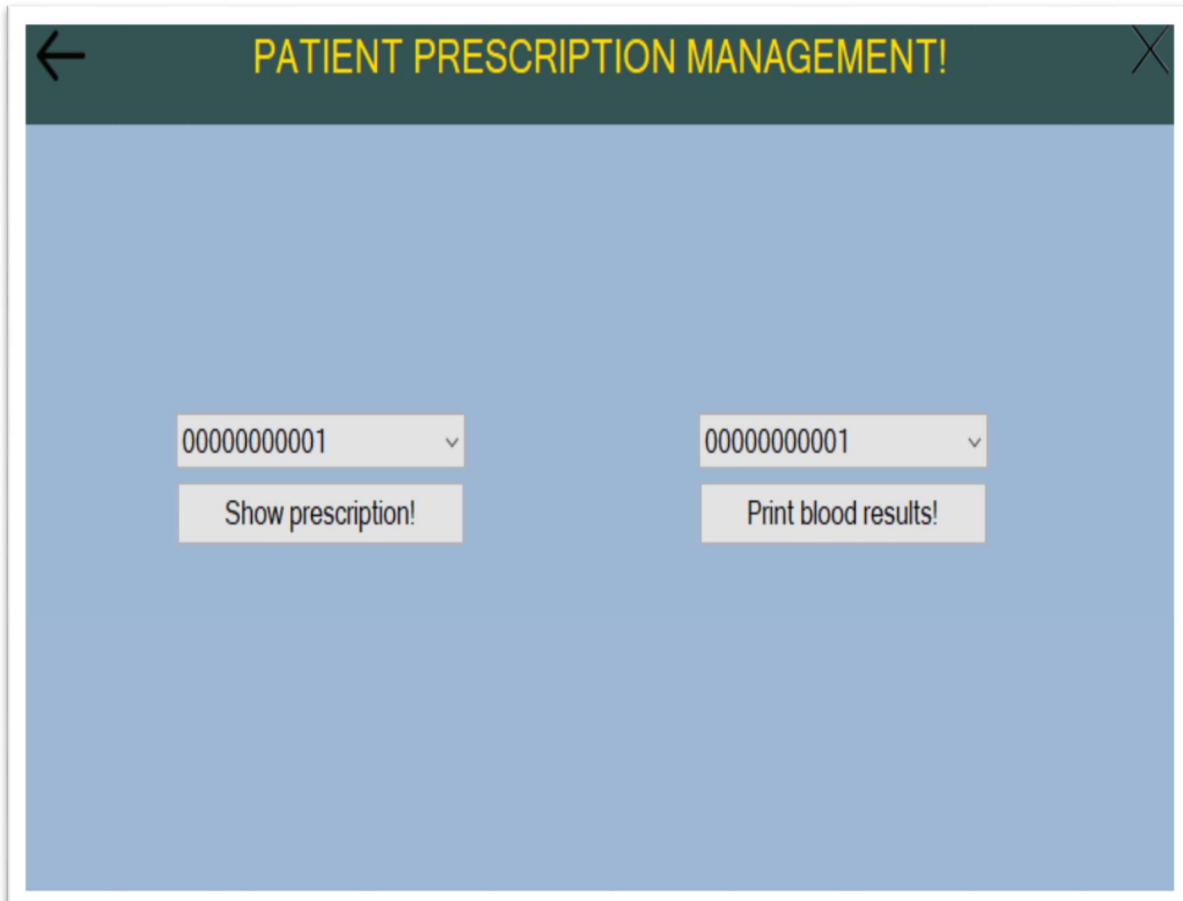
Αν επιθυμεί να αποθηκεύσει έναν ασθενή στο σύστημα, τότε θα πρέπει πρώτα να έχει ήδη αποθηκεύσει τις βασικές πληροφορίες του συγκεκριμένου ασθενή χρησιμοποιώντας την κατηγορία **Patients** ώστε να υπάρχει τον ΑΜΚΑ του ήδη στη βάση. Αφού το κάνει αυτό, τότε θα πρέπει να συμπληρώσει όλα τα κουτιά στη κατηγορία **Blood Section** και στη συνέχεια να πατήσει το κουμπί '**Save**'.

Αν επιθυμεί να ενημερώσει τις πληροφορίες κάποιου ασθενή που υπάρχει ήδη στο σύστημα αίματος, τότε πρέπει υποχρεωτικά να έχει συμπληρώσει το κουτί ΑΜΚΑ με το σωστό αριθμό του ασθενή που υπάρχει στη βάση, και στη συνέχεια να συμπληρώσει από 1 έως όλα τα υπόλοιπα κουτιά των οποίων τις πληροφορίες επιθυμεί να ενημερώσει. Αφού υλοποιήσει σωστά τις παραπάνω διαδικασίες, πατάει το κουμπί '**Update**' για να τις αποθηκεύσει στο σύστημα.

Αν τέλος επιθυμεί να διαγράψει έναν ήδη υπάρχον ασθενή από το σύστημα, τότε το μόνο που έχει να κάνει είναι να πληκτρολογήσει στο κουτί ΑΜΚΑ τον αριθμό φορολογικού μητρώου του ασθενή του οποίου θέλει να διαγράψει από το σύστημα και στη συνέχεια να πατήσει το κουμπί '**Delete**'. **ΠΡΟΣΟΧΗ!!!** Εδώ θα πρέπει να σημειωθεί ότι αυτή η διαδικασία δεν αναιρείται και αν γίνει από τη κατηγορία **BloodSection** τότε διαγράφει οποιαδήποτε πληροφορία σχετίζεται με το ΑΜΚΑ του συγκεκριμένου ασθενή μόνο από το πίνακα της βάσης δεδομένων μας που σχετίζεται με τις αιματολογικές εξετάσεις.

### 3.1.4.Prescriptions

Πατώντας το εικονίδιο της συγκεκριμένης κατηγορίας ο ιατρός μεταφέρεται στη παρακάτω φόρμα (βλ. Εικόνα 6):



Εικόνα 6 Prescription form

Εδώ ο ιατρός έχει δύο επιλογές. Πατώντας το κουμπί **'Show prescription!'** θα εμφανιστεί ένα document με σχετικές πληροφορίες για τον ασθενή που έχει το ΑΜΚΑ το οποίο διαλέξαμε στο σχετικό dropdown το οποίο θα έχει αυτή τη μορφή (βλ. Εικόνα 7):

**\*\*\*Prescription\*\*\***

AMKA :00000000001  
Name :aas  
Address :asssas  
Age :22  
Gender :Male  
Phone :1212121212

**Medicines prescribed:**

assa

**Doctor Chris T.**

**Εικόνα 7 Prescription details**

Ο ιατρός αυτό μπορεί να το εκτυπώσει για να το δώσει στον ασθενή σαν συνταγογράφηση των φαρμάκων που του προτείνει να αγοράσει για να αντιμετωπίσει το θέμα υγείας που έχει.

Έπειτα, πατώντας το κουμπί **'Print blood results!'** θα εμφανιστεί ένα document με σχετικές πληροφορίες για τις αιματολογικές εξετάσεις του ασθενή που έχει το ΑΦΜ το οποίο διαλέξαμε στο σχετικό dropdown το οποίο θα έχει αυτή τη μορφή (βλ. Εικόνα 8,9):



**Εικόνα 8 Blood results details**



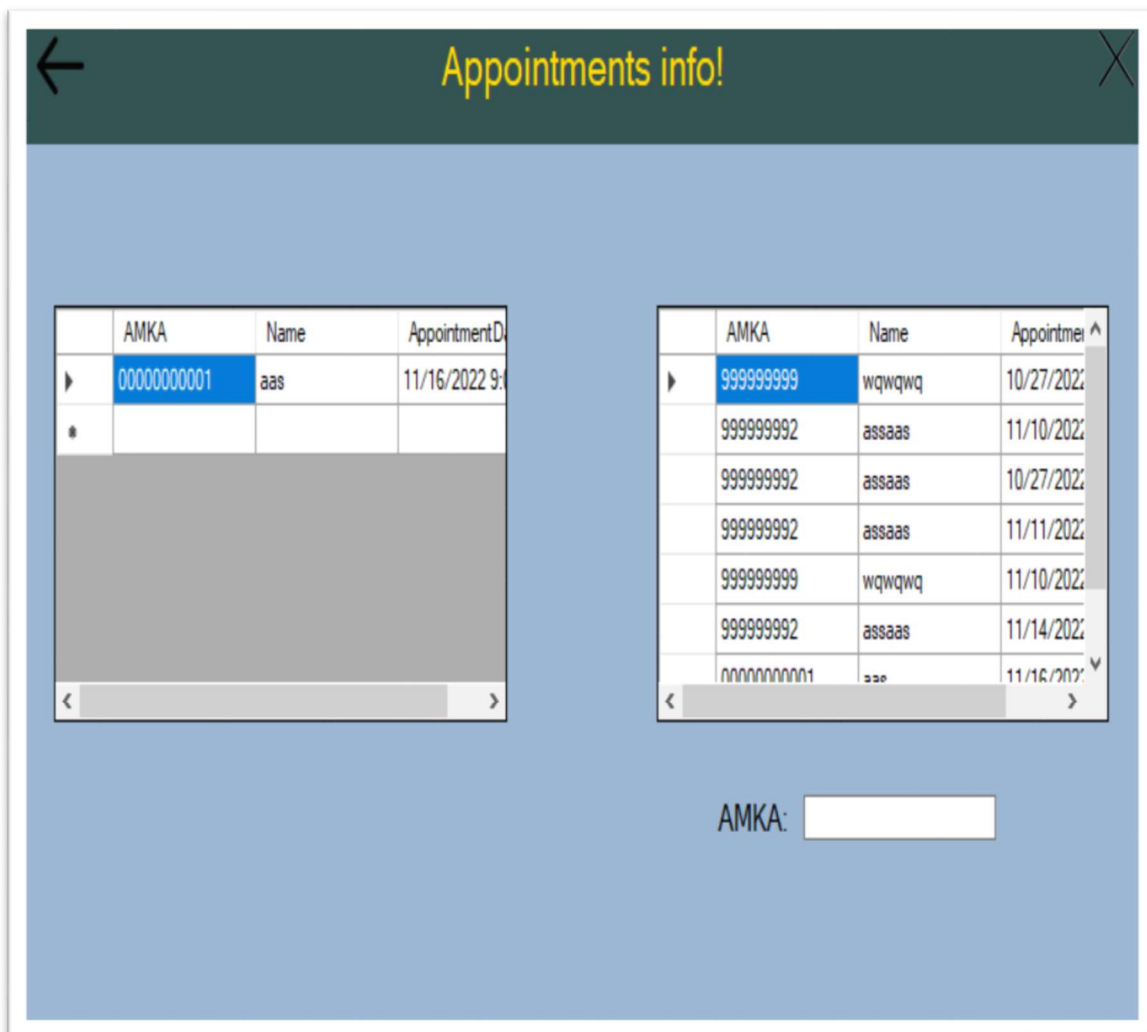
<b>RDW: 13</b>	Average: 4.00 - 10.50
<b>Platelets</b>	
<b>PLT: 14</b>	Average: 4.00 - 10.50
<b>PCT: 15</b>	Average: 4.00 - 10.50
<b>MPV: 16</b>	Average: 4.00 - 10.50
<b>PDW: 17</b>	Average: 4.00 - 10.50

**Εικόνα 9 Blood results details part 2**

Ο ιατρός μπορεί να εκτυπώσει το document και να το παραδώσει στον ασθενή ώστε να μπορεί κι αυτός να το προσκομίσει εκεί που χρειάζεται.

### 3.1.5. Appointments

Πατώντας το εικονίδιο της συγκεκριμένης κατηγορίας ο ιατρός μεταφέρεται στη παρακάτω φόρμα (βλ. Εικόνα 10):



Εικόνα 10 Appointments form

Στο παραπάνω σχήμα ο ιατρός στον αριστερό πίνακα βλέπει τα εκκρεμή ραντεβού του, ενώ στο δεξιό μπορεί να δει για κάθε ασθενή του που υπάρχει στο σύστημα όλα τα παλαιότερα και εκκρεμή ραντεβού που έχει κλείσει στο ιατρείο, και στο κάτω δεξιά κουτί μπορεί να κάνει αναζήτηση ενός συγκεκριμένου AMKA για να δει πληροφορίες των ραντεβού σχετικά με αυτό.

### 3.2.Web application

Ανοίγοντας την εφαρμογή μας παρουσιάζεται η παρακάτω σελίδα (βλ. Εικόνα 11):



Εικόνα 11 Web Login

Εδώ οι χρήστες, ασθενείς ή ιατρός, καλούνται να βάλουν τα στοιχεία τους στα κουτιά **Username-Password** και αν γίνει σωστή ταυτοποίηση με τα δεδομένα στη βάση, πατώντας το κουμπί **Login** θα μεταφερθούν στις σχετικές σελίδες. Ο ιατρός θα μεταφερθεί στη σελίδα **Menu** ενώ οι απλοί χρήστες (ασθενείς) θα μεταφερθούν στη σελίδα **Appointments**.

Πατώντας το κουμπί **Sign up!** οι χρήστες θα μεταφερθούν στη παρακάτω σελίδα (βλ. Εικόνα 12):



The image shows a web sign-up form titled "Sign Up!" overlaid on a blue background featuring a stethoscope and various medical icons. The form fields are as follows:

- Full Name**: Enter your name!
- Age**: Enter your age!
- Phone Number**: Enter your name!
- Gender**: Mr. / Ms.
- Height**: Enter your height!
- Weight**: Enter your weight!
- Blood Type**: O, A, B, AB
- AMKA**: Enter your AMKA!
- Address**: Enter your address!
- Login Credentials**:
  - Username**: Enter your username!
  - Password**: Enter your password!
  - Confirm Password**: Enter your password!

A green "Sign up!" button is located at the bottom of the form.

**Εικόνα 12 Web Sign Up**

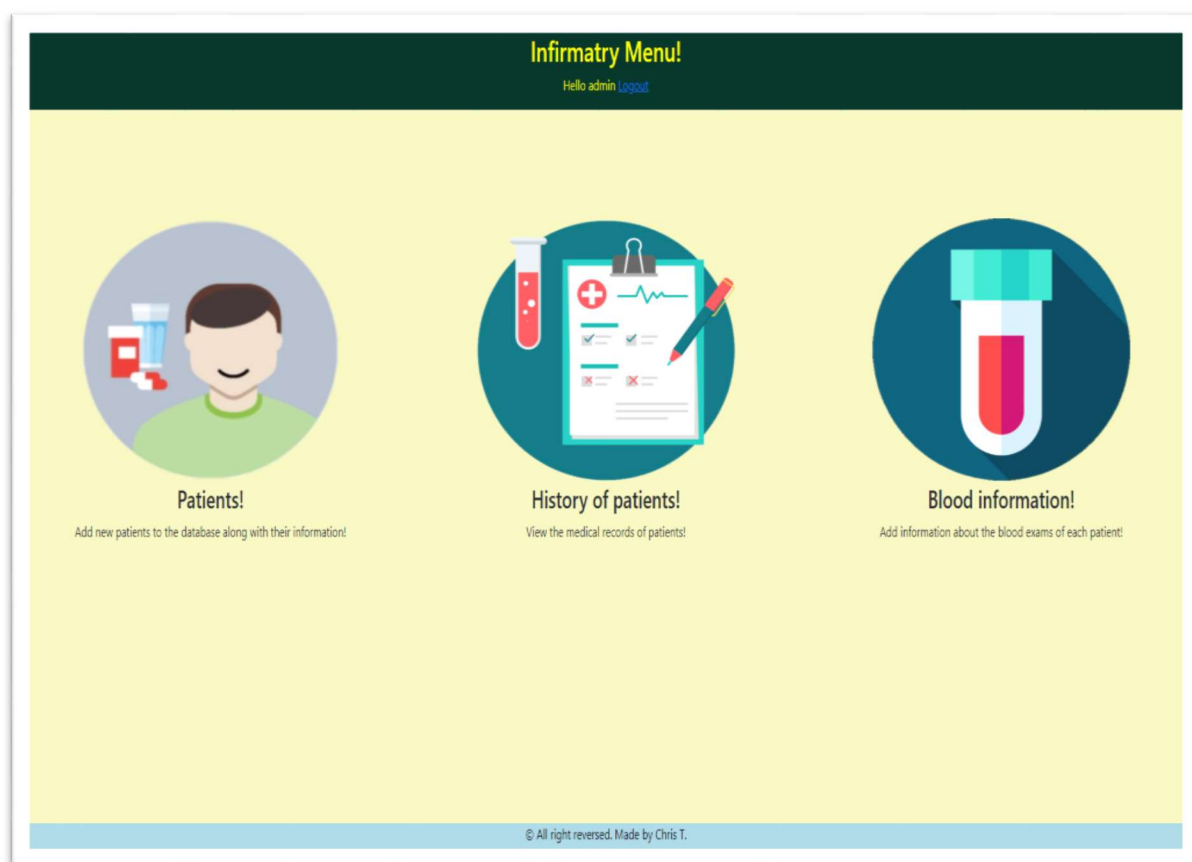
Αυτή η σελίδα προορίζεται για τους ασθενείς. Εδώ οι χρήστες καλούνται να πληκτρολογήσουν τα στοιχεία τους στα σχετικά κουτιά, και έπειτα να δημιουργήσουν έναν λογαριασμό για να αποκτήσουν πρόσβαση στις υπηρεσίες της εφαρμογής.

**ΠΡΟΣΟΧΗ!!!** Να σημειωθεί ότι αν κάποιος ασθενής έχει πρώτα επισκεφθεί το ιατρό, και ο τελευταίος έκανε καταχώρηση των στοιχείων του ασθενή μέσω της εφαρμογής του ιατρού, τότε ο χρήστης για να φτιάξει λογαριασμό πρέπει μόνο να γράψει σωστά τον AMKA του με τον οποίο τον καταχώρησε ο ιατρός, και έπειτα να πληκτρολογήσει το **Username – Password – Confirm Password** για να δημιουργήσει έναν λογαριασμό και να αποκτήσει πρόσβαση στις επιπλέον υπηρεσίες.

Στη συνέχεια θα αναλύσουμε τις σελίδες και λειτουργίες στις οποίες αποκτά πρόσβαση κάθε τύπος χρήστη (ιατρός, ασθενής).

### 3.2.1. Ιατρός

Αν ο ιατρός πληκτρολόγησε σωστά τα στοιχεία του, τότε θα μεταφερθεί στην παρακάτω σελίδα (βλ. Εικόνα 13):



Εικόνα 13 Web Menu

Υπάρχουν 3 κατηγορίες τις οποίες μπορεί να επιλέξει ο ιατρός τις οποίες θα αναλύσουμε στη συνέχεια:

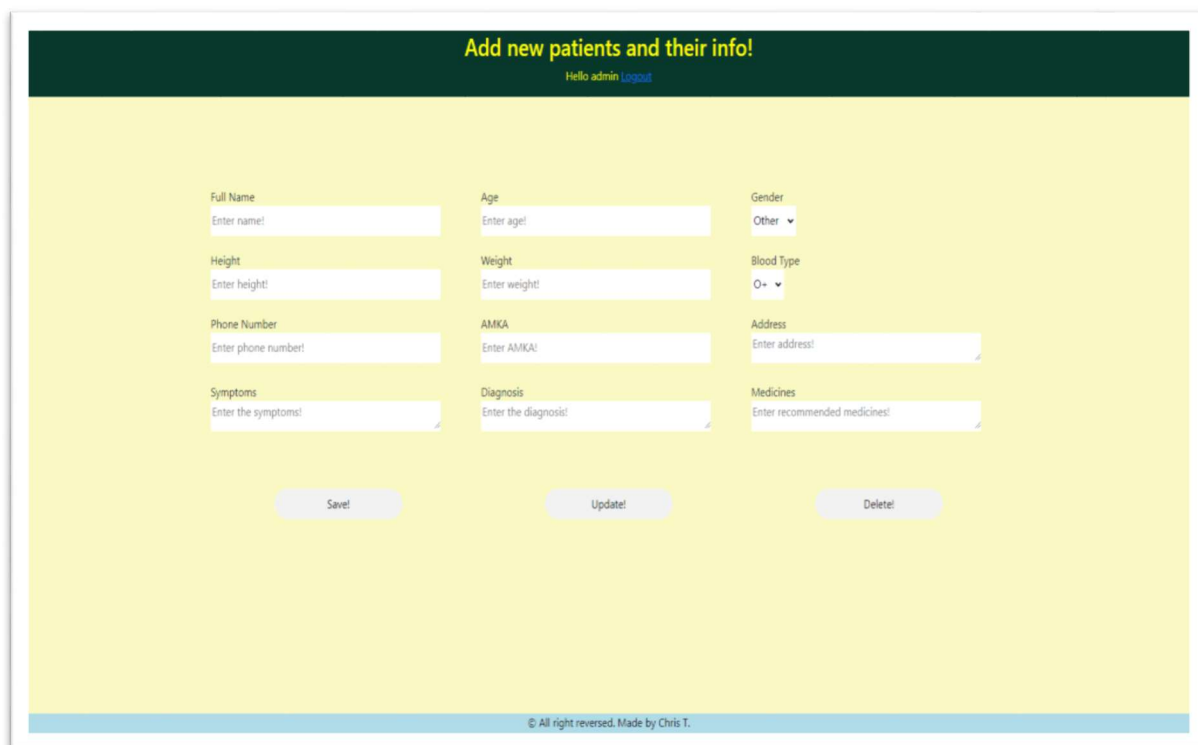
- Patients!
- History of patients!
- Blood information!

Επιπλέον σε κάθε κατηγορία της σελίδας πατώντας το [στη πάνω δεξιά μεριά](#) κάθε σελίδας αποσυνδεόμαστε από τον λογαριασμό μας και επιστρέφουμε στη σελίδα **Login**.

Παρακάτω περιγράφονται οι λειτουργίες κάθε κατηγορίας του μενού:

## Patients

Πατώντας στο εικονίδιο της συγκεκριμένης κατηγορίας ο ιατρός μεταφέρεται στη παρακάτω σελίδα (βλ. Εικόνα 14):



**Add new patients and their info!**  
Hello admin [Logout](#)

Full Name Enter name!	Age Enter age!	Gender Other ▾
Height Enter height!	Weight Enter weight!	Blood Type O+ ▾
Phone Number Enter phone number!	AMKA Enter AMKA!	Address Enter address!
Symptoms Enter the symptoms!	Diagnosis Enter the diagnosis!	Medicines Enter recommended medicines!

Save! Update! Delete!

© All right reversed. Made by Chris T.

**Εικόνα 14 Web Patients**

Στη συνέχεια ο ιατρός καλείται να συμπληρώσει έναν συγκεκριμένο αριθμό από τα διαθέσιμα textboxes που δίνονται ανάλογα με το σκοπό του.

Αν επιθυμεί να αποθηκεύσει ένα νέο ασθενή στο σύστημα τότε θα πρέπει να συμπληρώσει όλα τα κουτιά και στη συνέχεια να πατήσει το κουμπί **'Save'**.

Αν επιθυμεί να ενημερώσει τις πληροφορίες κάποιου ασθενή που υπάρχει ήδη στο σύστημα τότε πρέπει υποχρεωτικά να έχει συμπληρώσει το κουτί ΑΜΚΑ με το σωστό αριθμό του ασθενή που υπάρχει στη βάση, και στη συνέχεια να συμπληρώσει από 1 έως όλα τα υπόλοιπα κουτιά των οποίων τις πληροφορίες επιθυμεί να ενημερώσει.

Αφού υλοποιήσει σωστά τις παραπάνω διαδικασίες, πατάει το κουμπί **'Update'** για να τις αποθηκεύσει στο σύστημα.

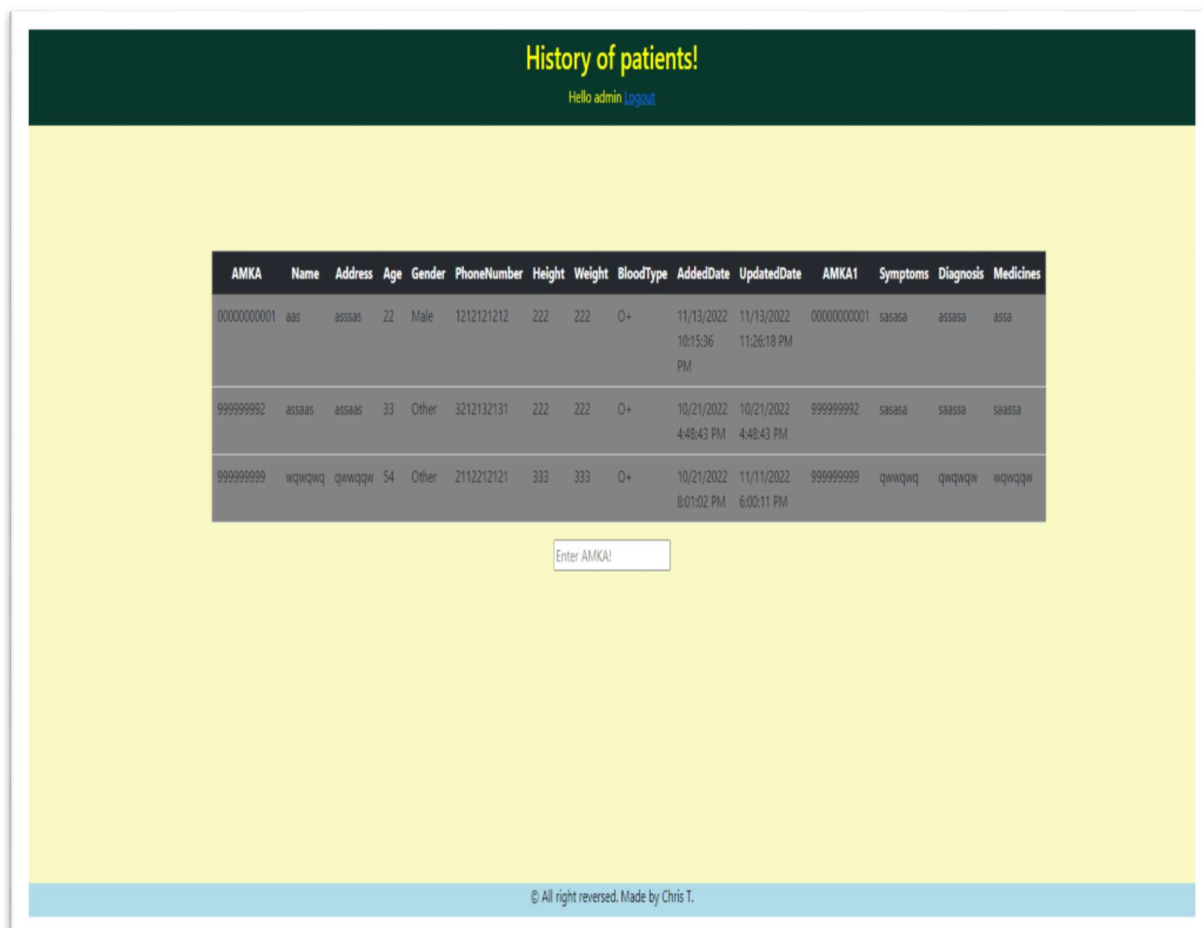
Αν τέλος επιθυμεί να διαγράψει έναν ήδη υπάρχον ασθενή από το σύστημα, τότε το μόνο που έχει να κάνει είναι να πληκτρολογήσει στο κουτί ΑΜΚΑ τον αριθμό φορολογικού μητρώου του ασθενή του οποίου θέλει να διαγράψει από το σύστημα και στη συνέχεια να πατήσει το κουμπί **'Delete'**. **ΠΡΟΣΟΧΗ!!!** Εδώ θα πρέπει να σημειωθεί ότι αυτή η διαδικασία δεν αναιρείται και διαγράφει πλήρως οποιαδήποτε πληροφορία σχετίζεται με τον ΑΜΚΑ του συγκεκριμένου ασθενή από κάθε πίνακα της βάσης δεδομένων μας.

Παρακάτω αναλύουμε τι πληροφορίες πρέπει να περιέχει κάθε textbox:

- Full Name**: ονοματεπώνυμο
- Age**: ηλικία
- Gender**: φύλο
- PhoneNumber**: αριθμός τηλεφώνου
- Height**: ύψος
- Weight**: βάρος
- Blood Type**: τύπος αίματος
- ΑΜΚΑ**: αριθμός μητρώου κοινωνικής ασφάλισης
- Address**: διεύθυνση κατοικίας
- Symptoms**: συμπτώματα
- Diagnosis**: διάγνωση του ιατρού
- Medicines**: φάρμακα που προτείνει ο ιατρός

## History of patients!

Πατώντας το εικονίδιο της συγκεκριμένης κατηγορίας ο ιατρός μεταφέρεται στη παρακάτω φόρμα (βλ. Εικόνα 15):



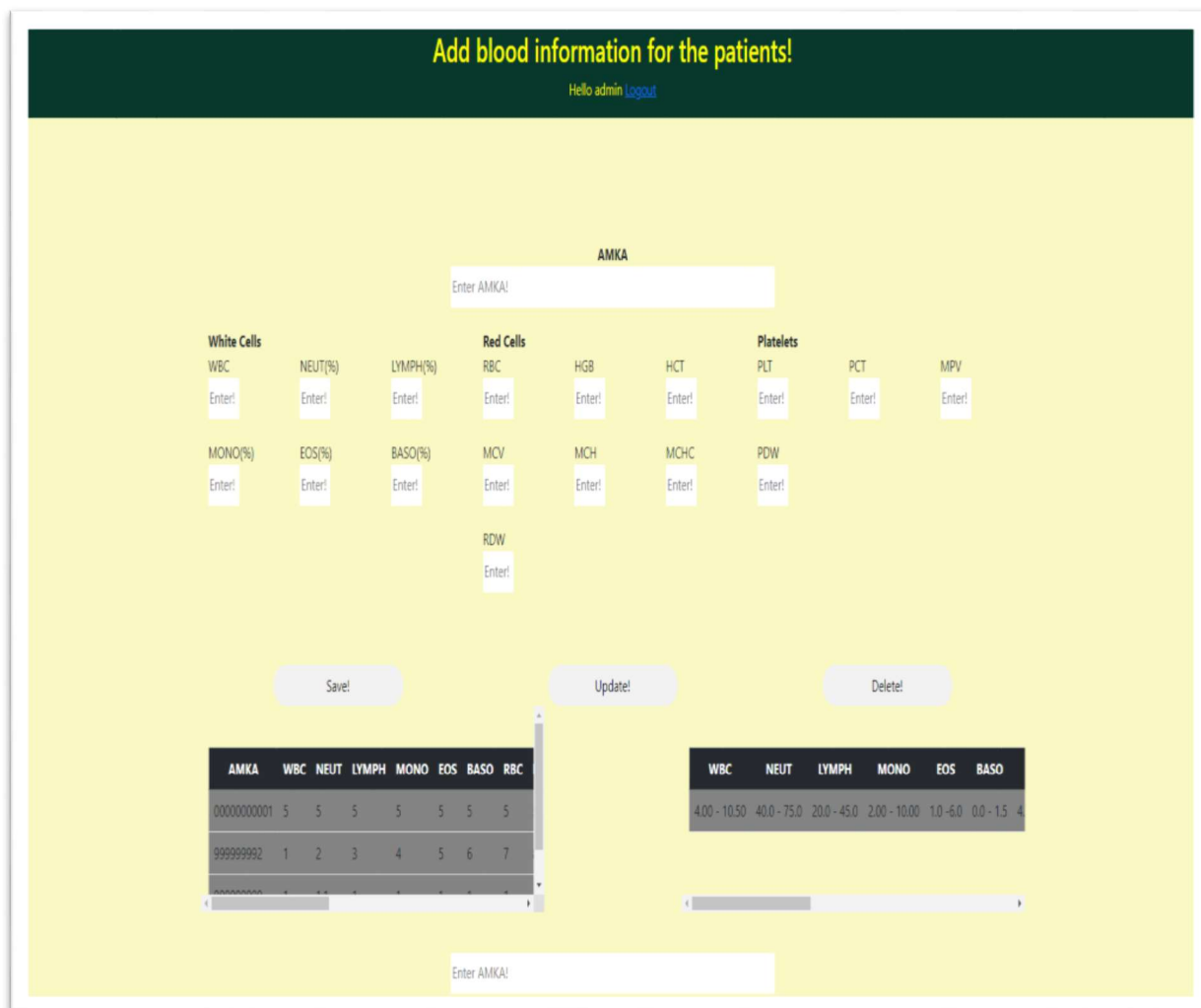
Εικόνα 15 Web History of patients

Εδώ ο ιατρός έχει τη δυνατότητα να δει τις βασικές πληροφορίες για κάθε ασθενή που έχει καταχωρήσει στο σύστημα, και χρησιμοποιώντας το textboχστη κάτω πλευρά της φόρμας μπορεί να κάνει αναζήτηση ενός συγκεκριμένου AMKA για να δει αμέσως τις πληροφορίες για τον ασθενή με το AMKA που πληκτρολόγησε στο κουτί.



### Blood information!

Πατώντας το εικονίδιο της συγκεκριμένης κατηγορίας ο ιατρός μεταφέρεται στη παρακάτω φόρμα (βλ. Εικόνα 16):



Εικόνα 16 Web Blood section

Ο κάτω αριστερά πίνακας δείχνει όλους τους ασθενείς για τους οποίους ο ιατρός έχει καταχωρήσει εξετάσεις αίματος στο σύστημα, και παρόμοια με το κουτί που βρίσκεται κάτω από τον συγκεκριμένο πίνακα ο ιατρός πληκτρολογώντας έναν AMKA ενός ασθενή που υπάρχει στη βάση μπορεί να δει τις πληροφορίες για το αίμα του. Ο πίνακας κάτω δεξιά δείχνει τις τιμές τις οποίες πρέπει να έχει κάθε στοιχείο του αίματος για να θεωρείται ‘εντός ορίων’. Ο ιατρός επίσης καλείται να συμπληρώσει έναν συγκεκριμένο αριθμό από τα διαθέσιμα textboxes που δίνονται ανάλογα με το σκοπό του.

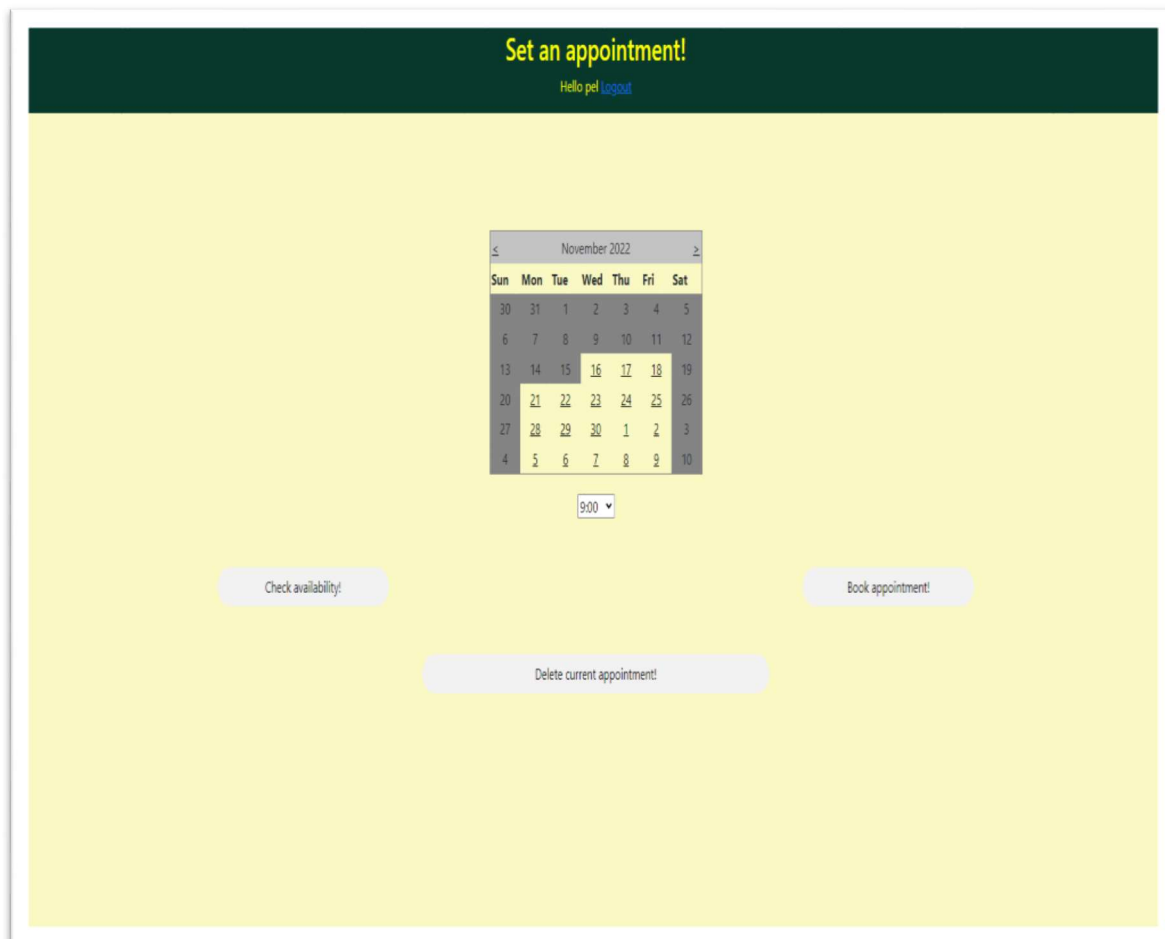
Αν επιθυμεί να αποθηκεύσει έναν ασθενή στο σύστημα, τότε θα πρέπει πρώτα να έχει ήδη αποθηκεύσει τις βασικές πληροφορίες του συγκεκριμένου ασθενή χρησιμοποιώντας την κατηγορία **Patients** ώστε να υπάρχει τον ΑΜΚΑ του ήδη στη βάση. Αφού το κάνει αυτό, τότε θα πρέπει να συμπληρώσει όλα τα κουτιά στη κατηγορία **Blood information!** και στη συνέχεια να πατήσει το κουμπί **'Save'**.

Αν επιθυμεί να ενημερώσει τις πληροφορίες κάποιου ασθενή που υπάρχει ήδη στο σύστημα αίματος, τότε πρέπει υποχρεωτικά να έχει συμπληρώσει το κουτί ΑΜΚΑ με το σωστό αριθμό του ασθενή που υπάρχει στη βάση, και στη συνέχεια να συμπληρώσει από 1 έως όλα τα υπόλοιπα κουτιά των οποίων τις πληροφορίες επιθυμεί να ενημερώσει. Αφού υλοποιήσει σωστά τις παραπάνω διαδικασίες, πατάει το κουμπί **'Update'** για να τις αποθηκεύσει στο σύστημα.

Αν τέλος επιθυμεί να διαγράψει έναν ήδη υπάρχον ασθενή από το σύστημα, τότε το μόνο που έχει να κάνει είναι να πληκτρολογήσει στο κουτί ΑΜΚΑ τον αριθμό φορολογικού μητρώου του ασθενή του οποίου θέλει να διαγράψει από το σύστημα και στη συνέχεια να πατήσει το κουμπί **'Delete'**. **ΠΡΟΣΟΧΗ!!!** Εδώ θα πρέπει να σημειωθεί ότι αυτή η διαδικασία δεν αναιρείται και αν γίνει από τη κατηγορία **Blood information!** τότε διαγράφει οποιαδήποτε πληροφορία σχετίζεται με το ΑΜΚΑ του συγκεκριμένου ασθενή μόνο από το πίνακα της βάσης δεδομένων μας που σχετίζεται με τις αιματολογικές εξετάσεις.

### 3.2.2.Ασθενείς

Αν ο ασθενής πληκτρολόγησε σωστά τα στοιχεία του, τότε θα μεταφερθεί στην παρακάτω σελίδα (βλ. Εικόνα 17):



Εικόνα 17 Web Appointments

Εδώ ο ασθενής έχει στη διάθεσή του ένα ημερολόγιο στο οποίο μπορεί να ψάξει και να επιλέξει ημερομηνία στην οποία επιθυμεί να κλείσει ραντεβού για επίσκεψη στο ιατρείο. Οι ημερομηνίες οι οποίες βρίσκονται σε γκρι πλαίσιο είναι μη διαθέσιμες για προγραμματισμό ενός ραντεβού.

Ο ασθενής αφού επιλέξει μια διαθέσιμη ημερομηνία, διαλέγει την ώρα που τον βολεύει χρησιμοποιώντας το dropdown που βρίσκεται ακριβώς κάτω από το ημερολόγιο. Έπειτα, πατώντας το κουμπί Check a vailability! ο χρήστης έχει τη δυνατότητα να ελέγξει αν υπάρχει διαθέσιμο ραντεβού για την ώρα που διάλεξε και στη συνέχεια του παρουσιάζεται σχετικό μήνυμα. Αν το μήνυμα δείξει ότι υπάρχει διαθέσιμο ραντεβού για την ώρα που διάλεξε ο

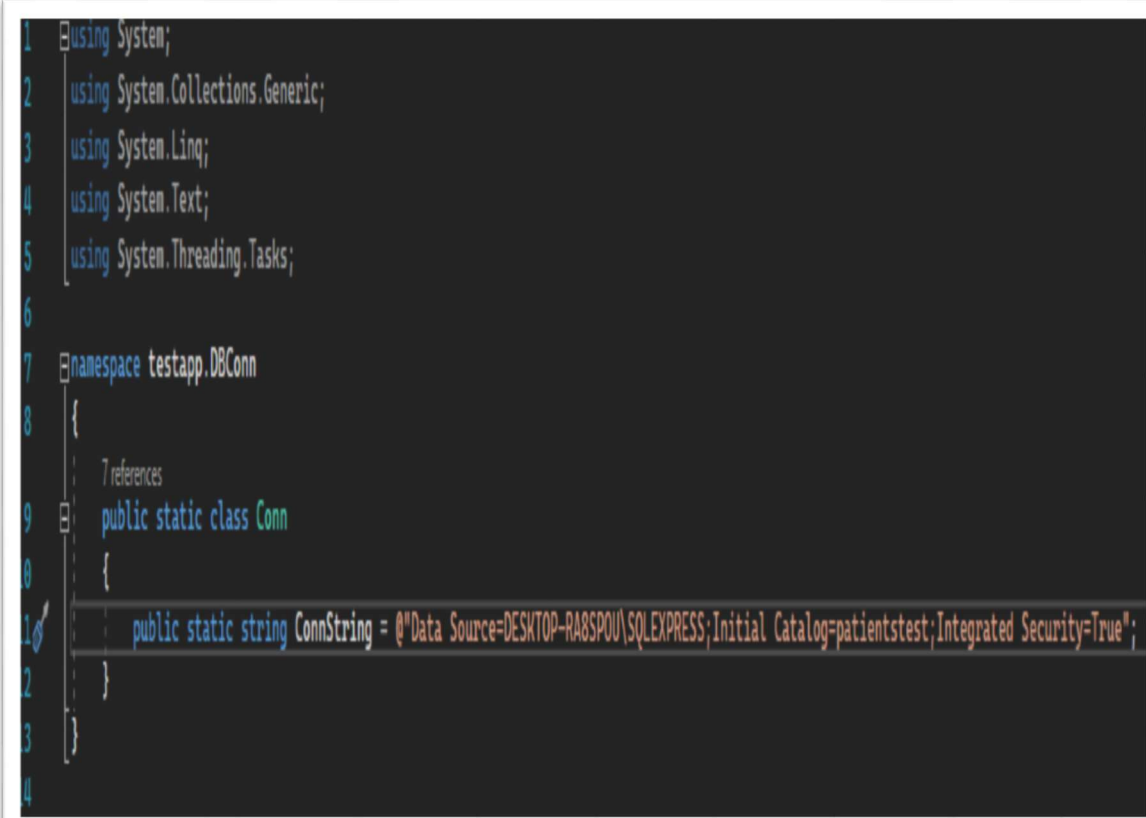
χρήστης, τότε πατώντας το κουμπί Book Appointment! μπορεί να καταχωρήσει και να κλείσει το ραντεβού του στη βάση.

Αν ο χρήστης έχει ήδη κλείσει κάποιο ραντεβού τότε θα του είναι διαθέσιμο και το κουμπί **Delete current appointment!** Αν ο χρήστης επιθυμεί να ακυρώσει το ενεργό ραντεβού του τότε πατώντας αυτό το κουμπί μπορεί να το ακυρώσει και στη συνέχεια να επιλέξει ένα νέο ραντεβού.

## 4.ΑΝΑΛΥΣΗ ΤΟΥ ΚΩΔΙΚΑ

### 4.1.Desktop application

Με την κλάση **Conn.cs** δημιουργούμε τη σύνδεση με τη βάση δεδομένων μας, και την επικαλούμαστε σε όλες τις φόρμες της εφαρμογής. (βλ. Εικόνα 18)

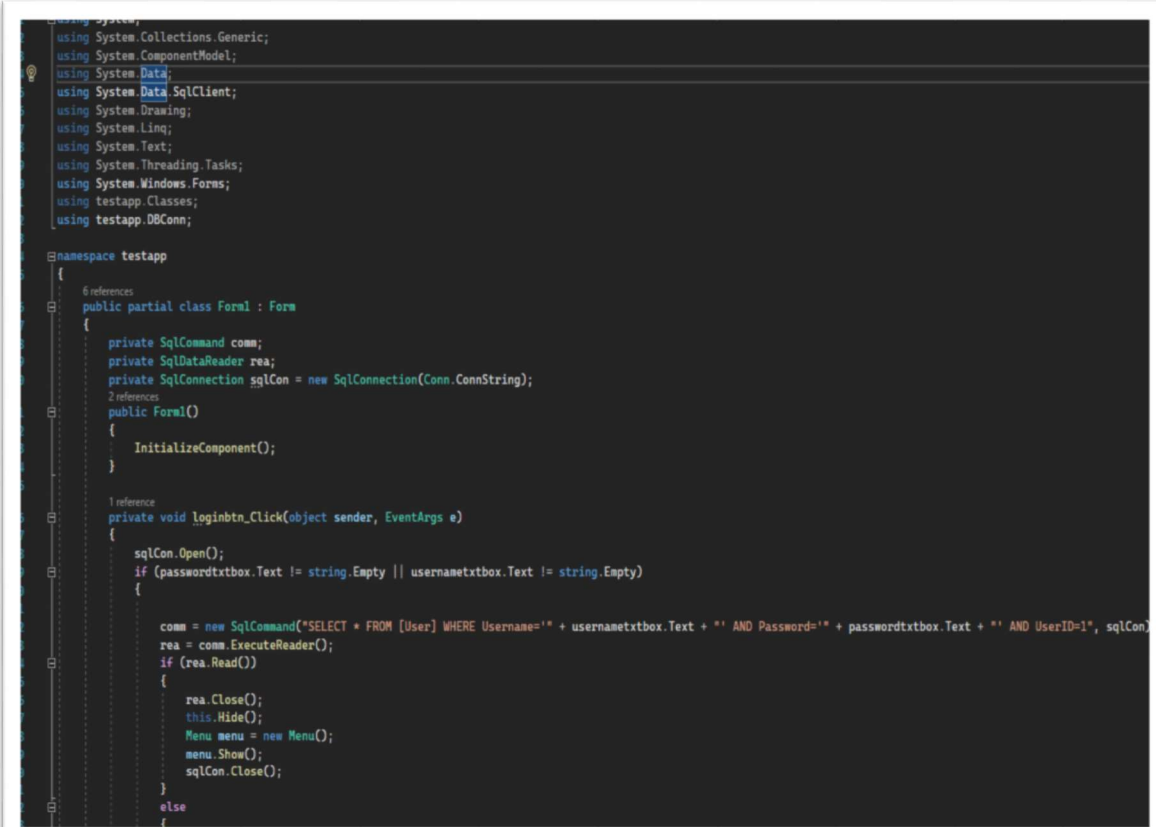


```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace testapp.DBConn
8 {
9     public static class Conn
10    {
11        public static string ConnString = @"Data Source=DESKTOP-RA8SPOU\SQLEXPRESS;Initial Catalog=patientstest;Integrated Security=True";
12    }
13 }
14
```

Εικόνα 18 Conn.cs code

### 4.1.1. Form1

Στη λειτουργία `loginbtn_Click` αφού πατήσουμε το κουμπί login, θα γίνει σύνδεση με τη βάση δεδομένων μας και ελέγχεται αν τα στοιχεία που έδωσε ο ιατρός είναι ίδια με τα στοιχεία που υπάρχουν στη βάση. Αν είναι σωστά τότε μεταφέρεται στη φόρμα Menu, αλλιώς αν κάτι πάει λάθος εμφανίζονται σχετικά μηνύματα. Επιπλέον σε όλες τις φόρμες πατώντας το εικονίδιο 'X' με τη λειτουργία `exitpicbox_Click` τερματίζουμε την εφαρμογή. (βλ. Εικόνα 19,20)



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using testapp.Classes;
using testapp.DBConn;

namespace testapp
{
    6 references
    public partial class Form1 : Form
    {
        private SqlCommand comm;
        private SqlDataReader rea;
        private SqlConnection sqlCon = new SqlConnection(Conn.ConnString);
        2 references
        public Form1()
        {
            InitializeComponent();
        }

        1 reference
        private void loginbtn_Click(object sender, EventArgs e)
        {
            sqlCon.Open();
            if (passwordtxtbox.Text != string.Empty || usernametxtbox.Text != string.Empty)
            {
                comm = new SqlCommand("SELECT * FROM [User] WHERE Username='" + usernametxtbox.Text + "' AND Password='" + passwordtxtbox.Text + "' AND UserID=1", sqlCon);
                rea = comm.ExecuteReader();
                if (rea.Read())
                {
                    rea.Close();
                    this.Hide();
                    Menu menu = new Menu();
                    menu.Show();
                    sqlCon.Close();
                }
            }
            else
            {
            }
        }
    }
}
```

Εικόνα 19 loginbtn\_Click code

```

43     {
44         rea.Close();
45         MessageBox.Show("No Account avilable with this username and password ", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
46         usernametxtbox.Text = "";
47         passwordtxtbox.Text = "";
48         usernametxtbox.Focus();
49         sqlCon.Close();
50     }
51 }
52 }
53 }
54 }
55 }
56 }
57 }
58 }
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```

Εικόνα 20 loginbtn\_Click code part 2

### 4.1.2.Menu

Στη συγκεκριμένη φόρμα κάθε εικόνα μας μεταφέρει σε διαφορετικές λειτουργίες της εφαρμογής αφού πατήσουμε πάνω τους. Επιπλέον με το βέλος που υπάρχει στην πάνω αριστερά μεριά κάθε φόρμας μεταφερόμαστε στη προηγούμενη λειτουργία όπου βρισκόμασταν. (βλ. Εικόνα 21,22)

```

11 namespace testapp
12 {
13     15 references
14     public partial class Menu : Form
15     {
16         6 references
17         public Menu()
18         {
19             InitializeComponent();
20         }
21     }
22     1 reference
23     private void patientspicbox_Click(object sender, EventArgs e)
24     {
25         this.Hide();
26         PatientForm patientform = new PatientForm();
27         patientform.Show();
28     }
29     1 reference
30     private void patienthistorypicbox_Click(object sender, EventArgs e)
31     {
32         this.Hide();
33         PatientHistory patienthistory = new PatientHistory();
34         patienthistory.Show();
35     }
36     1 reference
37     private void prescriptionpicbox_Click(object sender, EventArgs e)
38     {
39         this.Hide();
40         Prescription prescription = new Prescription();
41         prescription.Show();
42     }
43     1 reference
44     private void bloodpicbox_Click(object sender, EventArgs e)
45     {
46         this.Hide();
47         BloodForm bloodform = new BloodForm();
48         bloodform.Show();
49     }
50 }
51 }
52 }
53 }
54 }
55 }
56 }
57 }
58 }
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```

Εικόνα 21 Menu form code

```
48 | 1 reference  
49 | private void bckpicbox_Click(object sender, EventArgs e)  
50 | {  
51 |     this.Hide();  
52 |     Form1 form1 = new Form1();  
53 |     form1.Show();  
54 | }  
55 | 1 reference  
56 | private void exitpicbox_Click(object sender, EventArgs e)  
57 | {  
58 |     Application.Exit();  
59 | }  
60 | 1 reference  
61 | private void appointmentpic_Click(object sender, EventArgs e)  
62 | {  
63 |     this.Hide();  
64 |     Appointments appointments = new Appointments();  
65 |     appointments.Show();  
66 | }  
67 | }
```

Εικόνα 22 Menu form code part 2

### 4.1.3 Patient Form

English Key Press: επιτρέπει την πληκτρολόγηση μόνο αγγλικών γραμμάτων. (βλ. Εικόνα 23)

Letters Key Press: επιτρέπει την πληκτρολόγηση μόνο γραμμάτων. (βλ. Εικόνα 23)

Num Key Press: επιτρέπει την πληκτρολόγηση μόνο αριθμών (χρησιμοποιείται και σε άλλες φόρμες). (βλ. Εικόνα 24)

Patient Form\_Load: με τη φόρτωση της φόρμας δίνει σε κάθε πεδίο περιορισμούς ως προς τι μπορεί να πληκτρολογήσει ο χρήστης πάνω σε αυτό. (βλ. Εικόνα 24)

```

13 namespace Testapp
14 {
15     5 references
16     public partial class PatientForm : Form
17     {
18         private SqlCommand comm;
19         private SqlDataReader rea;
20         private SqlConnection sqlCon = new SqlConnection(Conn.ConnString);
21
22         1 reference
23         public PatientForm()
24         {
25             InitializeComponent();
26         }
27
28         4 references
29         private void EnglishKeyPress(object sender, KeyPressEventArgs e)
30         {
31             if (System.Text.Encoding.UTF8.GetByteCount(new char[] { e.KeyChar }) > 1)
32             {
33                 e.Handled = true;
34             }
35         }
36
37         1 reference
38         private void LettersKeyPress(object sender, KeyPressEventArgs e)
39         {
40             // επιτρέπει μόνο γράμματα και backspace
41             if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar) && (e.KeyChar != ' '))
42             {
43                 e.Handled = true;
44             }
45
46             // επιτρέπει το κείμενο να έχει μόνο ένα κενό
47             if ((e.KeyChar == ' ') && ((sender as TextBox).Text.IndexOf(' ') > -1))
48             {
49                 e.Handled = true;
50             }
51         }
52     }
53 }

```

Εικόνα 23 Press codes

```

49     private void NumKeyPress(object sender, KeyPressEventArgs e)
50     {
51         // επιτρέπει μόνο αριθμούς και backspace
52         if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))
53         {
54             e.Handled = true;
55         }
56     }
57
58     1 reference
59     private void PatientForm_Load(object sender, EventArgs e)
60     {
61         nametxtbox.KeyPress += LettersKeyPress;
62         nametxtbox.KeyPress -= NumKeyPress;
63         nametxtbox.KeyPress += EnglishKeyPress;
64         agetxtbox.KeyPress += NumKeyPress;
65         phonetxtbox.KeyPress += NumKeyPress;
66         heighttxtbox.KeyPress += NumKeyPress;
67         weighttxtbox.KeyPress += NumKeyPress;
68         amkatxtbox.KeyPress += NumKeyPress;
69         symptomsrtxtbox.KeyPress += EnglishKeyPress;
70         diagnosisirtxtbox.KeyPress += EnglishKeyPress;
71         medicinesrtxtbox.KeyPress += EnglishKeyPress;
72     }

```

Εικόνα 24 Press codes part 2



savebtn\_Click: κάνει τη σύνδεση με τη βάση δεδομένων, έπειτα ελέγχει αν όλα τα πεδία έχουν συμπληρωθεί, και στη συνέχεια κάνει τους απαραίτητους ελέγχους για να δει αν θα αποθηκεύσει τα δεδομένα στους κατάλληλους πίνακες της βάσης ή αν θα δείξει άλλα σχετικά μηνύματα. (βλ. Εικόνα 25,26)

```

73 private void savebtn_Click(object sender, EventArgs e)
74 {
75     sqlCon.Open();
76     comm = new SqlCommand();
77     comm.Connection = sqlCon;
78     comm.CommandType = CommandType.Text;
79     comm.CommandText = "select * from Patients where AMKA='" + amkatxtbox.Text + "'";
80     //έλεγχος αν υπάρχουν στοιχεία σε όλα τα πεδία
81     if (nametxtbox.Text == "" || ageetxtbox.Text == "" || genderbox.Text == "" || phonetxtbox.Text == "" || weighttxtbox.Text == "" || heighttxtbox.Text == "" || bloodcbo
82     {
83         comm.Dispose();
84         sqlCon.Close();
85         MessageBox.Show("All text boxes must be filled!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
86     }
87     else
88     {
89         SqlDataReader rea = comm.ExecuteReader();
90         //έλεγχος αν υπάρχει ήδη το ID που πληκτρολόγησε ο χρήστης
91         if (rea.HasRows)
92         {
93             rea.Close();
94             comm.Dispose();
95             sqlCon.Close();
96             MessageBox.Show("ID already exists!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
97         }
98         else
99         {
100             rea.Close();
101             comm.CommandText = "Insert into Patients values(@AMKA, @Name, @Address, @Age, @Gender, @PhoneNumber, @Height, @Weight, @BloodType, @AddedDate, @UpdatedDate, @Username)";
102             comm.Parameters.Add(new SqlParameter("@AMKA", amkatxtbox.Text));
103             comm.Parameters.Add(new SqlParameter("@Name", nametxtbox.Text));
104             comm.Parameters.Add(new SqlParameter("@Address", addressrtxtbox.Text));
105             comm.Parameters.Add(new SqlParameter("@Age", Convert.ToInt64(ageetxtbox.Text)));
106             comm.Parameters.Add(new SqlParameter("@Gender", genderbox.Text));
107             comm.Parameters.Add(new SqlParameter("@PhoneNumber", phonetxtbox.Text));
108             comm.Parameters.Add(new SqlParameter("@Height", Convert.ToInt64(heighttxtbox.Text)));
109             comm.Parameters.Add(new SqlParameter("@Weight", Convert.ToInt64(weighttxtbox.Text)));
110             comm.Parameters.Add(new SqlParameter("@BloodType", bloodcbox.Text));
111             comm.Parameters.Add(new SqlParameter("@AddedDate", DateTime.Now));
112             comm.Parameters.Add(new SqlParameter("@UpdatedDate", DateTime.Now));
113             comm.Parameters.Add(new SqlParameter("@Username", DBNull.Value));
114             comm.ExecuteNonQuery();
115             comm.Dispose();
116             sqlCon.Close();
117             sqlCon.Open();

```

Εικόνα 25 savebtn\_Click patients code app

```

117         sqlCon.Open();
118         comm = new SqlCommand();
119         comm.Connection = sqlCon;
120         comm.CommandType = CommandType.Text;
121         comm.CommandText = "Insert into Conclusions values(@AMKA,@Symptoms,@Diagnosis,@Medicines)";
122         comm.Parameters.Add(new SqlParameter("@AMKA", amkatxtbox.Text));
123         comm.Parameters.Add(new SqlParameter("@Symptoms", symptomsrtxtbox.Text));
124         comm.Parameters.Add(new SqlParameter("@Diagnosis", diagnosisrtxtbox.Text));
125         comm.Parameters.Add(new SqlParameter("@Medicines", medicinesrtxtbox.Text));
126         comm.ExecuteNonQuery();
127         comm.Dispose();
128         sqlCon.Close();
129         nametxtbox.Text = "";
130         agetxtbox.Text = "";
131         genderbox.Text = "";
132         phonetxtbox.Text = "";
133         heighttxtbox.Text = "";
134         weighttxtbox.Text = "";
135         bloodcbox.Text = "";
136         addressrtxtbox.Text = "";
137         amkatxtbox.Text = "";
138         symptomsrtxtbox.Text = "";
139         diagnosisrtxtbox.Text = "";
140         medicinesrtxtbox.Text = "";
141         MessageBox.Show("Information added!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
142     }
143 }
144

```

Εικόνα 26 savebtn\_Click patients code app part 2

updatebtn\_Click: κάνει τη σύνδεση με τη βάση δεδομένων, έπειτα ελέγχει αν όλα τα απαραίτητα πεδία έχουν συμπληρωθεί, και στη συνέχεια κάνει τους απαραίτητους ελέγχους για να δει αν θα αποθηκεύσει τα δεδομένα στους κατάλληλους πίνακες της βάσης ή αν θα δείξει άλλα σχετικά μηνύματα. (βλ. Εικόνα 27)

```

146 private void updatebtn_Click(object sender, EventArgs e)
147 {
148     sqlCon.Open();
149     comm = new SqlCommand();
150     comm.Connection = sqlCon;
151     comm.CommandType = CommandType.Text;
152     comm.CommandText = "select * from Patients where AMKA=" + amkatxtbox.Text + " ";
153     //ελέγχος αν υπάρχουν στοιχεία σε όλα τα πεδία
154     if (amkatxtbox.Text == "")
155     {
156         comm.Dispose();
157         sqlCon.Close();
158         MessageBox.Show("Left AMKA box must be filled with something!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
159     }
160     else if (nametxtbox.Text == "" && agetxtbox.Text == "" && genderbox.Text == "" && phonetxtbox.Text == "" && weighttxtbox.Text == "" && heighttxtbox.Text == "" && bloodcbox.Text == "" && addressrtxtbox.Text == "" && amkatxtbox.Text == "" && symptomsrtxtbox.Text == "" && diagnosisrtxtbox.Text == "" && medicinesrtxtbox.Text == "")
161     {
162         comm.Dispose();
163         sqlCon.Close();
164         MessageBox.Show("At least one text box must be filled!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
165     }
166     else
167     {
168         SqlDataReader rea = comm.ExecuteReader();
169         if (rea.HasRows)
170         {
171             rea.Close();
172             if (nametxtbox.Text != "")
173             {
174                 comm.CommandText = "Update Patients set Name=@Name where AMKA=@AMKA";
175                 comm.Parameters.Add(new SqlParameter("@AMKA", amkatxtbox.Text));
176                 comm.Parameters.Add(new SqlParameter("@Name", nametxtbox.Text));
177                 comm.ExecuteNonQuery();
178                 comm.Dispose();
179                 sqlCon.Close();
180                 sqlCon.Open();
181                 comm = new SqlCommand();
182                 comm.Connection = sqlCon;
183                 comm.CommandType = CommandType.Text;

```

Εικόνα 27 updatebtn\_Click patients code app

deletebtn\_Click: κάνει τη σύνδεση με τη βάση δεδομένων, έπειτα ελέγχει αν όλα τα απαραίτητα πεδία έχουν συμπληρωθεί, και στη συνέχεια κάνει τους απαραίτητους ελέγχους για να δει αν θα διαγράψει τα δεδομένα από όλους τους πίνακες της βάσης, οι οποίοι έχουν πληροφορίες σχετικές με το ΑΜΚΑ που θέλει να διαγράψει ο ιατρός, ή αν θα δείξει άλλα σχετικά μηνύματα. (βλ. Εικόνα 28,29)

```

345 private void deletebtn_Click(object sender, EventArgs e)
346 {
347     sqlCon.Open();
348     comm = new SqlCommand();
349     comm.Connection = sqlCon;
350     comm.CommandType = CommandType.Text;
351     comm.CommandText = "select * from Patients where ΑΜΚΑ=" + askatxtbox.Text + " ";
352     //έλεγχος αν υπάρχουν στοιχεία σε όλα τα πεδία
353     if (askatxtbox.Text == "")
354     {
355         comm.Dispose();
356         sqlCon.Close();
357         MessageBox.Show("Left ΑΜΚΑ box must be filled with something!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
358     }
359     else
360     {
361         SqlDataReader rea = comm.ExecuteReader();
362         //έλεγχος αν υπάρχει ήδη το ID που πληκτρολόγησε ο χρήστης
363         if (rea.HasRows)
364         {
365             rea.Close();
366             comm.CommandText = "Delete from Patients where ΑΜΚΑ=@ΑΜΚΑ";
367             comm.Parameters.Add(new SqlParameter("@ΑΜΚΑ", askatxtbox.Text));
368             comm.ExecuteNonQuery();
369             comm.Dispose();
370             sqlCon.Close();
371             sqlCon.Open();
372             comm = new SqlCommand();
373             comm.Connection = sqlCon;
374             comm.CommandType = CommandType.Text;
375             comm.CommandText = "Delete from [User] where ΑΜΚΑ=@ΑΜΚΑ";
376             comm.Parameters.Add(new SqlParameter("@ΑΜΚΑ", askatxtbox.Text));
377             comm.ExecuteNonQuery();
378             comm.Dispose();
379             sqlCon.Close();
380             sqlCon.Open();
381             comm = new SqlCommand();
382             comm.Connection = sqlCon;
383             comm.CommandType = CommandType.Text;
384             comm.CommandText = "Delete from BloodInfo where ΑΜΚΑ=@ΑΜΚΑ";
385             comm.Parameters.Add(new SqlParameter("@ΑΜΚΑ", askatxtbox.Text));
386             comm.ExecuteNonQuery();
387             comm.Dispose();
388             sqlCon.Close();

```

Εικόνα 28 deletebtn\_Click patients code app

```
87         sqlCon.Close();
88     sqlCon.Close();
89     sqlCon.Open();
90     comm = new SqlCommand();
91     comm.Connection = sqlCon;
92     comm.CommandType = CommandType.Text;
93     comm.CommandText = "Delete from Conclusions where ΑΜΚΑ=@ΑΜΚΑ";
94     comm.Parameters.Add(new SqlParameter("@ΑΜΚΑ", amkatxtbox.Text));
95     comm.ExecuteNonQuery();
96     comm.Dispose();
97     sqlCon.Close();
98     sqlCon.Open();
99     comm = new SqlCommand();
100    comm.Connection = sqlCon;
101    comm.CommandType = CommandType.Text;
102    comm.CommandText = "Delete from Appointments where ΑΜΚΑ=@ΑΜΚΑ";
103    comm.Parameters.Add(new SqlParameter("@ΑΜΚΑ", amkatxtbox.Text));
104    comm.ExecuteNonQuery();
105    comm.Dispose();
106    sqlCon.Close();
107    amkatxtbox.Text = "";
108    MessageBox.Show("Information deleted!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
109 }
110 else
111 {
112     rea.Close();
113     comm.Dispose();
114     sqlCon.Close();
115     MessageBox.Show("ΑΜΚΑ doesnt exist in database!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
116 }
117 }
118 }
```

Εικόνα 29 deletebtn\_Click patients code app part 2

#### 4.1.4. Patient History

Patient History\_Load: όταν ανοίγει η φόρμα εμφανίζει στο data Grid View1 τα δεδομένα που υπάρχουν στο συνδυασμό των πινάκων Patients-Conclusions της βάσης μας. (βλ. Εικόνα 30)

```
6 private void PatientHistory_Load(object sender, EventArgs e)
7 {
8     textBox1.KeyPress += NumKeyPress;
9     comm = new SqlCommand();
10    comm.Connection = sqlCon;
11    comm.CommandType = CommandType.Text;
12    comm.CommandText = "select * from Patients inner join Conclusions on Patients.AMKA = Conclusions.AMKA";
13    sda = new SqlDataAdapter(comm);
14    ds = new DataSet();
15    sda.Fill(ds);
16
17    dataGridView1.DataSource = ds.Tables[0];
18 }
19
```

Εικόνα 30 PatientHistory\_Load code app

textBox1\_TextChanged: κάνει αναζήτηση στο συνδυασμό πινάκων Patients-Conclusionτων AMKA που μοιάζουν με τα δεδομένα που πληκτρολογεί ο χρήστης στο πεδίο αναζήτησης.

(βλ. Εικόνα 31)

```
private void textBox1_TextChanged(object sender, EventArgs e)
{
    if (textBox1.Text != "")
    {
        string AMKA = textBox1.Text;
        comm = new SqlCommand();
        comm.Connection = sqlCon;
        comm.CommandType = CommandType.Text;
        comm.CommandText = "select * from Patients, Conclusions where Patients.AMKA = Conclusions.AMKA and Patients.AMKA Like '" + AMKA + "%' and Conclusions.AMKA Like '"
        sda = new SqlDataAdapter(comm);
        ds = new DataSet();
        sda.Fill(ds);

        dataGridView1.DataSource = ds.Tables[0];
    }
    else
    {
        LoadData();
    }
}
```

Εικόνα 31 textBox1\_TextChanged code app

LoadData: επαναφέρει τον πίνακα δεδομένων στην αρχική του κατάσταση αν το πεδίο αναζήτησης είναι κενό (βλ. Εικόνα 32)

```
51 private void LoadData()
52 {
53     comm = new SqlCommand();
54     comm.Connection = sqlCon;
55     comm.CommandType = CommandType.Text;
56     comm.CommandText = "select * from Patients inner join Conclusions on Patients.AMKA = Conclusions.AMKA";
57     sda = new SqlDataAdapter(comm);
58     ds = new DataSet();
59     sda.Fill(ds);
60
61     dataGridView1.DataSource = ds.Tables[0];
62 }
```

Εικόνα 32 LoadData code app

## Blood Form

NumDotKeyPress: επιτρέπει την πληκτρολόγηση μόνο αριθμών και το σύμβολο της τελείας μόνο 1 φορά, χωρίς αυτό να είναι στην 1<sup>η</sup> θέση του πεδίου. (βλ. Εικόνα 33)

```
53 private void NumDotKeyPress(object sender, KeyPressEventArgs e)
54 {
55     // επιτρέπει μόνο αριθμούς και backspace
56     if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) &&
57         (e.KeyChar != '.'))
58     {
59         e.Handled = true;
60     }
61     if ((e.KeyChar == '.') && ((sender as TextBox).Text.Length == 0))
62     {
63         e.Handled = true;
64     }
65     // επιτρέπει το κείμενο να έχει μόνο μία τελεία
66     if ((e.KeyChar == '.') && ((sender as TextBox).Text.IndexOf('.') > -1 ))
67     {
68         e.Handled = true;
69     }
70 }
```

Εικόνα 33 NumDotKeyPress code app



BloodForm\_Load: όταν ανοίγει η φόρμα εμφανίζει στον πίνακα dataGridView1 τα δεδομένα που υπάρχουν στον πίνακα Blood Infoτης βάσης μας, ενώ εμφανίζει στο πίνακα data Grid View2 τα δεδομένα ου υπάρχουν στο πίνακα Blood Average της βάσης μας. (βλ. Εικόνα 34)

```

72 private void BloodForm_Load(object sender, EventArgs e)
73 {
74     amkatxtbox.KeyPress += NumKeyPress;
75     amkav2txtbox.KeyPress += NumKeyPress;
76     wbctxtbox.KeyPress += NumDotKeyPress;
77     neuttxtbox.KeyPress += NumDotKeyPress;
78     Lymphtxtbox.KeyPress += NumDotKeyPress;
79     monotxtbox.KeyPress += NumDotKeyPress;
80     eastxtbox.KeyPress += NumDotKeyPress;
81     basotxtbox.KeyPress += NumDotKeyPress;
82     rbctxtbox.KeyPress += NumDotKeyPress;
83     hgbtxtbox.KeyPress += NumDotKeyPress;
84     hcttxtbox.KeyPress += NumDotKeyPress;
85     mcvtxtbox.KeyPress += NumDotKeyPress;
86     mchtxtbox.KeyPress += NumDotKeyPress;
87     mchctxtbox.KeyPress += NumDotKeyPress;
88     rdwtxtbox.KeyPress += NumDotKeyPress;
89     plttxtbox.KeyPress += NumDotKeyPress;
90     pcttxtbox.KeyPress += NumDotKeyPress;
91     mpvtxtbox.KeyPress += NumDotKeyPress;
92     pdwtxtbox.KeyPress += NumDotKeyPress;
93     comm = new SqlCommand();
94     comm.Connection = sqlCon;
95     comm.CommandType = CommandType.Text;
96     comm.CommandText = "select * from BloodAverage";
97     sda = new SqlDataAdapter(comm);
98     ds = new DataSet();
99     sda.Fill(ds);
100
101     dataGridView2.DataSource = ds.Tables[0];
102
103     comm = new SqlCommand();
104     comm.Connection = sqlCon;
105     comm.CommandType = CommandType.Text;
106     comm.CommandText = "select * from BloodInfo";
107     sda = new SqlDataAdapter(comm);
108     ds = new DataSet();
109     sda.Fill(ds);
110
111     dataGridView1.DataSource = ds.Tables[0];
112

```

Εικόνα 34 BloodForm\_Load code app

savebtn\_Click: κάνει τη σύνδεση με τη βάση δεδομένων, έπειτα ελέγχει αν όλα τα πεδία έχουν συμπληρωθεί, και στη συνέχεια κάνει τους απαραίτητους ελέγχους για να δει αν θα αποθηκεύσει τα δεδομένα στους κατάλληλους πίνακες της βάσης ή αν θα δείξει άλλα σχετικά μηνύματα. (βλ. Εικόνα 35,36,37)

```

38 {
39     sqlCon.Open();
40     comm = new SqlCommand();
41     comm.Connection = sqlCon;
42     comm.CommandType = CommandType.Text;
43     comm.CommandText = "select * from Patients where ΑΜΚΑ='" + amkatxtbox.Text + "'";
44     //έλεγχος αν υπάρχουν στοιχεία σε όλα τα πεδία
45     if (amkatxtbox.Text == "" || wbcxtxtbox.Text == "" || neuttxtbox.Text == "" || lymphtxtbox.Text == "" || monotxtbox.Text == "" || eosstxtbox.Text == "" || basotxtbox.Text == "" || rbctxtbox.Text == "" || hgbtxtbox.Text == "" || hcttxtbox.Text == "" || mcvtxtbox.Text == "" || mchtxtbox.Text == "" || rdwtxtbox.Text == "" || plttxtbox.Text == "" || pcttxtbox.Text == "" || mpvtxtbox.Text == "" || pdwtxtbox.Text == "")
46     {
47         comm.Dispose();
48         sqlCon.Close();
49         MessageBox.Show("All text boxes must be filled!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
50     }
51     else
52     {
53         SqlDataReader rea = comm.ExecuteReader();
54         //έλεγχος αν υπάρχει ήδη το ID που πληκτρολόγησε ο χρήστης
55         if (rea.HasRows)
56         {
57             rea.Close();
58             comm = new SqlCommand();
59             comm.Connection = sqlCon;
60             comm.CommandType = CommandType.Text;
61             comm.CommandText = "select * from BloodInfo where ΑΜΚΑ='" + amkatxtbox.Text + "'";
62             rea = comm.ExecuteReader();
63             if (rea.HasRows)
64             {
65                 rea.Close();
66                 comm.Dispose();
67                 sqlCon.Close();
68                 MessageBox.Show("ΑΜΚΑ already exists!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
69             }
70             else
71             {
72                 rea.Close();
73                 comm.Dispose();
74                 sqlCon.Close();
75                 sqlCon.Open();
76                 comm = new SqlCommand();
77                 comm.Connection = sqlCon;
78                 comm.CommandType = CommandType.Text;
79                 comm.CommandText = "Insert into BloodInfo values(@ΑΜΚΑ, @WBC, @NEUT, @LYMPH, @MONO, @EOS, @BASO, @RBC, @HGB, @HCT, @MCV, @MCH, @MCHC, @RDW, @PLT, @PCT, @MPV, @PDW, @ADDED_DATE, @UPDATED_DATE)";
80                 comm.Parameters.AddWithValue("@ΑΜΚΑ", amkatxtbox.Text);
81                 comm.Parameters.AddWithValue("@WBC", Convert.ToDecimal(wbcxtxtbox.Text));
82                 comm.Parameters.AddWithValue("@NEUT", Convert.ToDecimal(neuttxtbox.Text));
83                 comm.Parameters.AddWithValue("@LYMPH", Convert.ToDecimal(lymphtxtbox.Text));
84                 comm.Parameters.AddWithValue("@MONO", Convert.ToDecimal(monotxtbox.Text));
85                 comm.Parameters.AddWithValue("@EOS", Convert.ToDecimal(eosstxtbox.Text));
86                 comm.Parameters.AddWithValue("@BASO", Convert.ToDecimal(basotxtbox.Text));
87                 comm.Parameters.AddWithValue("@RBC", Convert.ToDecimal(rbctxtbox.Text));
88                 comm.Parameters.AddWithValue("@HGB", Convert.ToDecimal(hgbtxtbox.Text));
89                 comm.Parameters.AddWithValue("@HCT", Convert.ToDecimal(hcttxtbox.Text));
90                 comm.Parameters.AddWithValue("@MCV", Convert.ToDecimal(mcvtxtbox.Text));
91                 comm.Parameters.AddWithValue("@MCH", Convert.ToDecimal(mchtxtbox.Text));
92                 comm.Parameters.AddWithValue("@MCHC", Convert.ToDecimal(mchtxtbox.Text));
93                 comm.Parameters.AddWithValue("@RDW", Convert.ToDecimal(rdwtxtbox.Text));
94                 comm.Parameters.AddWithValue("@PLT", Convert.ToDecimal(plttxtbox.Text));
95                 comm.Parameters.AddWithValue("@PCT", Convert.ToDecimal(pcttxtbox.Text));
96                 comm.Parameters.AddWithValue("@MPV", Convert.ToDecimal(mpvtxtbox.Text));
97                 comm.Parameters.AddWithValue("@PDW", Convert.ToDecimal(pdwtxtbox.Text));
98                 comm.Parameters.AddWithValue("@ADDED_DATE", DateTime.Now);
99                 comm.Parameters.AddWithValue("@UPDATED_DATE", DateTime.Now);
100                comm.ExecuteNonQuery();
101                comm.Dispose();
102                sqlCon.Close();
103                wbcxtxtbox.Text = "";
104                neuttxtbox.Text = "";
105                lymphtxtbox.Text = "";
106                monotxtbox.Text = "";
107                eosstxtbox.Text = "";
108                basotxtbox.Text = "";
109                rbctxtbox.Text = "";
110                hgbtxtbox.Text = "";
111                hcttxtbox.Text = "";
112                mcvtxtbox.Text = "";
113                mchtxtbox.Text = "";
114                rdwtxtbox.Text = "";
115                plttxtbox.Text = "";
116                pcttxtbox.Text = "";
117                mpvtxtbox.Text = "";
118                pdwtxtbox.Text = "";
119                LoadData();
120                MessageBox.Show("Information added!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
121            }
122        }
123    }

```

Εικόνα 35 savebtn\_Click blood code app

```

71     comm.Parameters.AddWithValue("@WBC", Convert.ToDecimal(wbcxtxtbox.Text));
72     comm.Parameters.AddWithValue("@NEUT", Convert.ToDecimal(neuttxtbox.Text));
73     comm.Parameters.AddWithValue("@LYMPH", Convert.ToDecimal(lymphtxtbox.Text));
74     comm.Parameters.AddWithValue("@MONO", Convert.ToDecimal(monotxtbox.Text));
75     comm.Parameters.AddWithValue("@EOS", Convert.ToDecimal(eosstxtbox.Text));
76     comm.Parameters.AddWithValue("@BASO", Convert.ToDecimal(basotxtbox.Text));
77     comm.Parameters.AddWithValue("@RBC", Convert.ToDecimal(rbctxtbox.Text));
78     comm.Parameters.AddWithValue("@HGB", Convert.ToDecimal(hgbtxtbox.Text));
79     comm.Parameters.AddWithValue("@HCT", Convert.ToDecimal(hcttxtbox.Text));
80     comm.Parameters.AddWithValue("@MCV", Convert.ToDecimal(mcvtxtbox.Text));
81     comm.Parameters.AddWithValue("@MCH", Convert.ToDecimal(mchtxtbox.Text));
82     comm.Parameters.AddWithValue("@MCHC", Convert.ToDecimal(mchtxtbox.Text));
83     comm.Parameters.AddWithValue("@RDW", Convert.ToDecimal(rdwtxtbox.Text));
84     comm.Parameters.AddWithValue("@PLT", Convert.ToDecimal(plttxtbox.Text));
85     comm.Parameters.AddWithValue("@PCT", Convert.ToDecimal(pcttxtbox.Text));
86     comm.Parameters.AddWithValue("@MPV", Convert.ToDecimal(mpvtxtbox.Text));
87     comm.Parameters.AddWithValue("@PDW", Convert.ToDecimal(pdwtxtbox.Text));
88     comm.Parameters.AddWithValue("@ADDED_DATE", DateTime.Now);
89     comm.Parameters.AddWithValue("@UPDATED_DATE", DateTime.Now);
90     comm.ExecuteNonQuery();
91     comm.Dispose();
92     sqlCon.Close();
93     wbcxtxtbox.Text = "";
94     neuttxtbox.Text = "";
95     lymphtxtbox.Text = "";
96     monotxtbox.Text = "";
97     eosstxtbox.Text = "";
98     basotxtbox.Text = "";
99     rbctxtbox.Text = "";
100    hgbtxtbox.Text = "";
101    hcttxtbox.Text = "";
102    mcvtxtbox.Text = "";
103    mchtxtbox.Text = "";
104    rdwtxtbox.Text = "";
105    plttxtbox.Text = "";
106    pcttxtbox.Text = "";
107    mpvtxtbox.Text = "";
108    pdwtxtbox.Text = "";
109    LoadData();
110    MessageBox.Show("Information added!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
111 }

```

Εικόνα 36 savebtn\_Click blood code app part 2



```

15         else
16         {
17             rea.Close();
18             comm.Dispose();
19             sqlCon.Close();
20             MessageBox.Show("AMKA doesnt exist!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
21         }
22     }

```

Εικόνα 37 savebtn\_Click blood code app part 3

updatebtn\_Click: κάνει τη σύνδεση με τη βάση δεδομένων, έπειτα ελέγχει αν όλα τα απαραίτητα πεδία έχουν συμπληρωθεί, και στη συνέχεια κάνει τους απαραίτητους ελέγχους για να δει αν θα αποθηκεύσει τα δεδομένα στους κατάλληλους πίνακες της βάσης ή αν θα δείξει άλλα σχετικά μηνύματα. (βλ. Εικόνα 38)

```

25     {
26         sqlCon.Open();
27         comm = new SqlCommand();
28         comm.Connection = sqlCon;
29         comm.CommandType = CommandType.Text;
30         comm.CommandText = "select * from BloodInfo where AMKA=" + amkatxtbox.Text + " ";
31         //έλεγχος αν υπάρχουν στοιχεία σε όλα τα πεδία
32         if (amkatxtbox.Text == "")
33         {
34             comm.Dispose();
35             sqlCon.Close();
36             MessageBox.Show("Left AMKA box must be filled with something!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
37         }
38     }
39     else if (wbctxtbox.Text == "" && neuttxtbox.Text == "" && lymphtxtbox.Text == "" && monotxtbox.Text == "" && eosxtxtbox.Text == "" && basotxtbox.Text == "" &&
40     {
41         comm.Dispose();
42         sqlCon.Close();
43         MessageBox.Show("At least one text box must be filled!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
44     }
45     else
46     {
47         SqlDataReader rea = comm.ExecuteReader();
48         if (rea.HasRows)
49         {
50             rea.Close();
51             if (wbctxtbox.Text != "")
52             {
53                 comm.CommandText = "Update BloodInfo set MBC=@MBC where AMKA=@AMKA";
54                 comm.Parameters.Add(new SqlParameter("@AMKA", amkatxtbox.Text));
55                 comm.Parameters.Add(new SqlParameter("@MBC", Convert.ToDecimal(wbctxtbox.Text)));
56                 comm.ExecuteNonQuery();
57                 comm.Dispose();
58                 sqlCon.Close();
59                 sqlCon.Open();
60                 comm = new SqlCommand();
61                 comm.Connection = sqlCon;
62                 comm.CommandType = CommandType.Text;
63             }
64             if (neuttxtbox.Text != "")
65             {
66                 comm.CommandText = "Update BloodInfo set NEUT=@NEUT where AMKA=@AMKA";
67                 comm.Parameters.Add(new SqlParameter("@AMKA", amkatxtbox.Text));
68                 comm.Parameters.Add(new SqlParameter("@NEUT", Convert.ToDecimal(neuttxtbox.Text)));

```

Εικόνα 38 updatebtn\_Click blood code app

deletebtn\_Click: κάνει τη σύνδεση με τη βάση δεδομένων, έπειτα ελέγχει αν όλα τα απαραίτητα πεδία έχουν συμπληρωθεί, και στη συνέχεια κάνει τους απαραίτητους ελέγχους

για να δει αν θα διαγράψει τα δεδομένα μόνο από το πίνακα Blood Info της βάσης. (βλ. Εικόνα 39)

```

609 private void deletebtn_Click(object sender, EventArgs e)
610 {
611     sqlCon.Open();
612     comm = new SqlCommand();
613     comm.Connection = sqlCon;
614     comm.CommandType = CommandType.Text;
615     comm.CommandText = "select * from BloodInfo where AMKA=" + amkatxtbox.Text + "";
616     //έλεγχος αν υπάρχουν στοιχεία σε όλα τα πεδία
617     if (amkatxtbox.Text == "")
618     {
619         comm.Dispose();
620         sqlCon.Close();
621         MessageBox.Show("AMKA box must be filled with something!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
622     }
623     else
624     {
625         SqlDataReader rea = comm.ExecuteReader();
626         //έλεγχος αν υπάρχει ήδη το ID που πληκτρολόγησε ο χρήστης
627         if (rea.HasRows)
628         {
629             rea.Close();
630             comm.CommandText = "Delete from BloodInfo where AMKA=@AMKA";
631             comm.Parameters.Add(new SqlParameter("@AMKA", amkatxtbox.Text));
632             comm.ExecuteNonQuery();
633             comm.Dispose();
634             sqlCon.Close();
635             amkatxtbox.Text = "";
636             LoadData();
637             MessageBox.Show("Information deleted!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
638         }
639         else
640         {
641             rea.Close();
642             comm.Dispose();
643             sqlCon.Close();
644             MessageBox.Show("AMKA doesnt exist in database!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
645         }
646     }
647 }
    
```

Εικόνα 39 deletebtn\_Click blood code app

amkav2txtbox\_TextChanged: κάνει αναζήτηση στο πίνακα BloodInfo των AMKA που μοιάζουν με τα δεδομένα που πληκτρολογεί ο χρήστης στο πεδίο αναζήτησης. (βλ. Εικόνα 40)

```

648 private void amkav2txtbox_TextChanged(object sender, EventArgs e)
649 {
650     if (amkav2txtbox.Text != "")
651     {
652         int AMKA = Convert.ToInt32(amkav2txtbox.Text);
653         comm = new SqlCommand();
654         comm.Connection = sqlCon;
655         comm.CommandType = CommandType.Text;
656         comm.CommandText = "select * from BloodInfo where BloodInfo.AMKA like '" + AMKA + "%'";
657         sda = new SqlDataAdapter(comm);
658         ds = new DataSet();
659         sda.Fill(ds);
660
661         dataGridView1.DataSource = ds.Tables[0];
662     }
663     else
664     {
665         LoadData();
666     }
667 }
    
```

Εικόνα 40 amkav2txtbox\_TextChanged code app

#### 4.1.5. Prescriptions

Load Patients: φορτώνει στα dropdowns τα διαθέσιμα ΑΜΚΑ που υπάρχουν στους πίνακες Patients και Blood Info. (βλ. Εικόνα 41)

```
34 private void LoadPatients()
35 {
36     using (SqlConnection sqlCon = new SqlConnection(Conn.ConnString))
37     {
38         try
39         {
40             string query = "select AMKA from Patients";
41             sda = new SqlDataAdapter(query, sqlCon);
42             sqlCon.Open();
43             ds = new DataSet();
44             sda.Fill(ds, "Patients");
45             amkacombobox.DisplayMember = "AMKA";
46             amkacombobox.ValueMember = "AMKA";
47             amkacombobox.DataSource = ds.Tables["Patients"];
48             sqlCon.Close();
49             query = "select AMKA from BloodInfo";
50             sda = new SqlDataAdapter(query, sqlCon);
51             sqlCon.Open();
52             ds = new DataSet();
53             sda.Fill(ds, "BloodInfo");
54             amkacombobox2.DisplayMember = "AMKA";
55             amkacombobox2.ValueMember = "AMKA";
56             amkacombobox2.DataSource = ds.Tables["BloodInfo"];
57         }
58         catch (Exception ex)
59         {
60             MessageBox.Show("Something went wrong! Please try again!");
61         }
62     }
63 }
64 }
```

Εικόνα 41 LoadPatients code app

OpenConnectionString: ανοίγει τη σύνδεση με τη βάση. (βλ. Εικόνα 42)

```
78 private SqlConnection OpenConnectionString(SqlConnection sqlCon)
79 {
80     if (sqlCon.State == ConnectionState.Open) sqlCon.Close();
81     sqlCon.Open();
82     return sqlCon;
83 }
```

Εικόνα 42 OpenConnection String code app

prescriptionbtn\_Click: εκτυπώνει το αρχείο με τις πληροφορίες της συνταγογράφησης για τον ασθενή που επιλέξαμε. (βλ. Εικόνα 43)

```

85 private void prescriptionbtn_Click(object sender, EventArgs e)
86 {
87
88     amka = amkacombox.SelectedValue.ToString();
89     if (printPreviewDialog1.ShowDialog() == DialogResult.OK && amka != "")
90     {
91         printDocument1.Print();
92     }
93 }

```

Εικόνα 43 prescriptionbtn\_Click code app

printDocument1\_PrintPage: παίρνει τις απαραίτητες πληροφορίες από τα δεδομένα των πινάκων Patients, Conclusions της βάσης μας και έπειτα τα εκτυπώνει σε σελίδες, σύμφωνα με το χρώμα, τη γραμματοσειρά και τη θέση στο χαρτί που ορίσαμε για κάθε δεδομένο. (βλ. Εικόνα 44)

```

96 private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
97 {
98     Patients patients = new Patients();
99     List<Conclusions> conclusions = new List<Conclusions>();
100
101     amka = amkacombox.SelectedValue.ToString();
102     if (amka != "")
103     {
104         sqlCon = OpenConnectionString(sqlCon);
105         string query = String.Format(@"select AMKA, Name, Address, Age, Gender, PhoneNumber from Patients where AMKA='{0}'", amka);
106         sda = new SqlDataAdapter(query, sqlCon);
107         DataTable dt = new DataTable();
108         sda.Fill(dt);
109         sqlCon.Close();
110
111         patients = this.GetPatientsInfo(dt);
112
113         sqlCon = OpenConnectionString(sqlCon);
114         query = String.Format(@"select Medicines from Conclusions where AMKA='{0}'", amka);
115         sda = new SqlDataAdapter(query, sqlCon);
116         dt = new DataTable();
117         sda.Fill(dt);
118         sqlCon.Close();
119
120         conclusions = this.GetMedicinesInfo(dt);
121
122         e.Graphics.DrawString("***Prescription***", new Font("Century", 22, FontStyle.Bold), Brushes.Red, new Point(200, 40));
123
124         e.Graphics.DrawString("AMKA : " + patients.AMKA, new Font("Arial", 14, FontStyle.Regular), Brushes.Black, new Point(120, 105));
125         e.Graphics.DrawString("Name : " + patients.Name, new Font("Arial", 14, FontStyle.Regular), Brushes.Black, new Point(120, 145));
126         e.Graphics.DrawString("Address : " + patients.Address, new Font("Arial", 14, FontStyle.Regular), Brushes.Black, new Point(120, 185));
127         e.Graphics.DrawString("Age : " + patients.Age, new Font("Arial", 14, FontStyle.Regular), Brushes.Black, new Point(120, 225));
128         e.Graphics.DrawString("Gender : " + patients.Gender, new Font("Arial", 14, FontStyle.Regular), Brushes.Black, new Point(120, 265));
129         e.Graphics.DrawString("Phone : " + patients.PhoneNumber, new Font("Arial", 14, FontStyle.Regular), Brushes.Black, new Point(120, 305));
130
131         e.Graphics.DrawString("Medicines prescribed:", new Font("Century", 18, FontStyle.Bold), Brushes.Red, new Point(200, 405));
132
133         foreach (var medicine in conclusions)
134         {
135             e.Graphics.DrawString(medicine.Medicines, new Font("Arial", 14, FontStyle.Regular), Brushes.Black, new Point(120, 445));
136         }
137
138         e.Graphics.DrawString("Doctor Chris T.", new Font("Century", 16, FontStyle.Bold), Brushes.Red, new Point(200, 1000));
139     }
140 }

```

Εικόνα 44 printDocument1\_PrintPage code app

GetPatientsInfo: παίρνει συγκεκριμένα δεδομένα από το πίνακα Patients της βάσης μας και τα αποθηκεύει σε κατάλληλες μεταβλητές για να χρησιμοποιηθούν από την εφαρμογή. (βλ.

Εικόνα 45)

```
141 private Patients GetPatientsInfo(DataTable dt)
142 {
143     Patients patients = (from rw in dt.AsEnumerable()
144                         select new Patients()
145                         {
146                             AMKA = Convert.ToString(rw["AMKA"]),
147                             Name = Convert.ToString(rw["Name"]),
148                             Address = Convert.ToString(rw["Address"]),
149                             Age = Convert.ToInt32(rw["Age"]),
150                             Gender = Convert.ToString(rw["Gender"]),
151                             PhoneNumber = Convert.ToString(rw["PhoneNumber"])
152                         }).ToList().FirstOrDefault();
153     return patients;
154 }
```

Εικόνα 45 GetPatientsInfo code app

GetMedicinesInfo: παίρνει συγκεκριμένα δεδομένα από το πίνακα Conclusion της βάσης μας και τα αποθηκεύει σε κατάλληλες μεταβλητές για να χρησιμοποιηθούν από την εφαρμογή.

(βλ. Εικόνα 46)

```
156 private List<Conclusions> GetMedicinesInfo(DataTable dt)
157 {
158     List<Conclusions> conclusions = (from rw in dt.AsEnumerable()
159                                     select new Conclusions()
160                                     {
161                                         Medicines = Convert.ToString(rw["Medicines"])
162                                     }).ToList();
163     return conclusions;
164 }
```

Εικόνα 46 GetMedicinesInfo code app



bloodbtn\_Click: εκτυπώνει το αρχείο με τις πληροφορίες των εξετάσεων αίματος για τον ασθενή που επιλέξαμε. (βλ. Εικόνα 47)

```

166     private void bloodbtn_Click(object sender, EventArgs e)
167     {
168         amka = amkacombobox2.SelectedValue.ToString();
169         i = 0;
170         if (printPreviewDialog2.ShowDialog() == DialogResult.OK && amka != "")
171         {
172             printDocument2.Print();
173         }
174     }

```

Εικόνα 47 bloodbtn\_Click code app

printDocument2\_PrintPage: παίρνει τις απαραίτητες πληροφορίες από τα δεδομένα των πινάκων Patients, BloodInfo της βάσης μας και έπειτα τα εκτυπώνει σε σελίδες, σύμφωνα με το χρώμα, τη γραμματοσειρά και τη θέση στο χαρτί που ορίσαμε για κάθε δεδομένο. Επιπλέον δημιουργεί έξτρα σελίδες αν τα δεδομένα δε χωράνε μόνο σε 1 σελίδα. (βλ. Εικόνα 48)

```

166     private void printDocument2_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
167     {
168         Patients patients = new Patients();
169         BloodInfo bloodinfo = new BloodInfo();
170
171         amka = amkacombobox2.SelectedValue.ToString();
172         m_Font = new Font("Arial", 14, FontStyle.Regular);
173         float lineHeight = m_Font.GetHeight(e.Graphics) + PAD;
174         float yLineTop = 48;
175         int startY = 18;
176         if (amka != "")
177         {
178             sqlCon = OpenConnectionString(sqlCon);
179             string query = String.Format(@"select AMKA, Name, Address, Age, Gender, PhoneNumber from Patients where AMKA='{0}'", amka);
180             sda = new SqlDataAdapter(query, sqlCon);
181             DataTable dt = new DataTable();
182             sda.Fill(dt);
183             sqlCon.Close();
184
185             patients = this.GetPatientsInfo(dt);
186
187             sqlCon = OpenConnectionString(sqlCon);
188             query = String.Format(@"select WBC, NEUT, LYMPH, MONO, EOS, BASO, RBC, HGB, HCT, MCV, MCH, MCHC, RDW, PLT, MPV, PDW from BloodInfo where AMKA='{0}'", amka);
189             sda = new SqlDataAdapter(query, sqlCon);
190             dt = new DataTable();
191             sda.Fill(dt);
192             sqlCon.Close();
193
194             bloodinfo = this.GetBloodInfo(dt);
195
196             if (i == 0)
197             {
198                 e.Graphics.DrawString("***Blood results***", new Font("Century", 22, FontStyle.Bold), Brushes.Red, new PointF(280, startY + yLineTop));
199                 yLineTop += lineHeight;
200                 e.Graphics.DrawString("AMKA : " + patients.AMKA, new Font("Arial", 14, FontStyle.Regular), Brushes.Black, new PointF(120, startY + yLineTop));
201                 yLineTop += lineHeight;
202                 e.Graphics.DrawString("Name : " + patients.Name, new Font("Arial", 14, FontStyle.Regular), Brushes.Black, new PointF(120, startY + yLineTop));
203                 yLineTop += lineHeight;
204                 e.Graphics.DrawString("Address : " + patients.Address, new Font("Arial", 14, FontStyle.Regular), Brushes.Black, new PointF(120, startY + yLineTop));
205                 yLineTop += lineHeight;
206                 e.Graphics.DrawString("Age : " + patients.Age, new Font("Arial", 14, FontStyle.Regular), Brushes.Black, new PointF(120, startY + yLineTop));
207                 yLineTop += lineHeight;
208                 e.Graphics.DrawString("Gender : " + patients.Gender, new Font("Arial", 14, FontStyle.Regular), Brushes.Black, new PointF(120, startY + yLineTop));
209                 yLineTop += lineHeight;

```

Εικόνα 48 printDocument2\_PrintPage code app

GetBloodInfo: παίρνει συγκεκριμένα δεδομένα από το πίνακα BloodInfo της βάσης μας και τα αποθηκεύει σε κατάλληλες μεταβλητές για να χρησιμοποιηθούν από την εφαρμογή. (βλ.

Εικόνα 49)

```

567 private BloodInfo GetBloodInfo(DataTable dt)
568 {
569     BloodInfo bloodinfo = (from rw in dt.AsEnumerable()
570     select new BloodInfo()
571     {
572         WBC = Convert.ToInt32(rw["WBC"]),
573         NEUT = Convert.ToInt32(rw["NEUT"]),
574         LYMPH = Convert.ToInt32(rw["LYMPH"]),
575         MONO = Convert.ToInt32(rw["MONO"]),
576         EOS = Convert.ToInt32(rw["EOS"]),
577         BASO = Convert.ToInt32(rw["BASO"]),
578         RBC = Convert.ToInt32(rw["RBC"]),
579         HGB = Convert.ToInt32(rw["HGB"]),
580         HCT = Convert.ToInt32(rw["HCT"]),
581         MCV = Convert.ToInt32(rw["MCV"]),
582         MCH = Convert.ToInt32(rw["MCH"]),
583         MCHC = Convert.ToInt32(rw["MCHC"]),
584         RDW = Convert.ToInt32(rw["RDW"]),
585         PLT = Convert.ToInt32(rw["PLT"]),
586         PCT = Convert.ToInt32(rw["PCT"]),
587         MPV = Convert.ToInt32(rw["MPV"]),
588         PDW = Convert.ToInt32(rw["PDW"]),
589     }).ToList().FirstOrDefault(); ;
590 }
591 }
592

```

Εικόνα 49 GetBloodInfo code app

#### 4.1.6. Appointments

Appointments\_Load: όταν ανοίγει η φόρμα εμφανίζει στον πίνακα dataGridView1 τα δεδομένα που υπάρχουν στον πίνακα Appointments της βάσης μας δείχνοντας μόνο τα ραντεβού που εκκρεμούν για ημερομηνίες μεγαλύτερες από τη σημερινή, ενώ εμφανίζει στο πίνακα dataGridView2 τα δεδομένα που υπάρχουν στο πίνακα Appointment της βάσης μας με όλα τα ραντεβού των ασθενών που υπήρξαν σε όλη τη διάρκεια λειτουργίας της εφαρμογής.

LoadData: επαναφέρει το πίνακα dataGridView2 στην αρχική του κατάσταση.

textBox1\_TextChanged: κάνει αναζήτηση στο πίνακα Appointments των ΑΜΚΑ που μοιάζουν με τα δεδομένα που πληκτρολογεί ο χρήστης στο πεδίο αναζήτησης.

## 4.2.Web application

Web.config: στο configuration δημιουργούμε τη σύνδεση με τη βάση δεδομένων μας και την επικαλούμαστε σε κάθε σελίδα της εφαρμογής. (βλ. Εικόνα 50)



```
<configuration>
  <connectionStrings>
    <add name="Conn" connectionString="Data Source=DESKTOP-RABSP0U\SQLEXPRESS;Initial Catalog=patientstest;Integrated Security=True" />
    <add name="patientstestConnectionString" connectionString="Data Source=DESKTOP-RABSP0U\SQLEXPRESS;Initial Catalog=patientstest;Integrated Security=True"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

Εικόνα 50 Web.config web app

<bodyonkeydown="return (event.keyCode!=13)">:

κώδικάς που χρησιμοποιείται στο σώμα όλων των σελίδων της εφαρμογής ώστε να μην μπορούν οι χρήστες να καταχωρήσουν δεδομένα πατώντας το πλήκτρο 'Enter'.

### 4.2.1.Login

Login.aspx.cs

loginbtn\_Click: ελέγχει τα στοιχεία που έδωσε ο χρήστης με αυτά που υπάρχουν στο πίνακα Users της βάσης δεδομένων και αν είναι σωστά τότε δίνει σχετικές τιμές σε 3 διαφορετικά Sessions και μεταφέρει το χρήστη στη κατάλληλη σελίδα ανάλογα με την τιμή που πήρε το sessionId. Αν πάει κάτι λάθος βγάζει σχετικό μήνυμα. (βλ. Εικόνα 51)



```

protected void Loginbtn_Click(object sender, EventArgs e)
{
    SqlConnection sqlCon = new SqlConnection(Con);
    if (sqlCon.State == ConnectionState.Closed)
    {
        sqlCon.Open();
    }
    SqlCommand cmd = new SqlCommand("SELECT * FROM [User] WHERE Username='" + txtuser.Text + "' AND Password='" + txtpassw.Text + "'", sqlCon);
    SqlDataReader dr = cmd.ExecuteReader();
    if (txtpassw.Text == string.Empty || txtuser.Text == string.Empty)
    {
        Response.Write("<script>alert('Please enter something in every field!');</script>");
    }
    else if (dr.HasRows)
    {
        while (dr.Read())
        {
            Response.Write("<script>alert('"+dr.GetValue(0).ToString()+"');</script>");
            Session["username"] = dr.GetValue(0).ToString();
            Session["amka"] = dr.GetValue(2).ToString();
            Session["userID"] = dr.GetValue(3).ToString();
        }
        if (Session["userID"].Equals("1"))
        {
            Response.Redirect("Menu.aspx");
        }
        else
        {
            Response.Redirect("Appointments.aspx");
        }
    }
    else
    {
        Response.Write("<script>alert('Invalid credentials!');</script>");
        txtuser.Text = "";
        txtpassw.Text = "";
        txtuser.Focus();
    }
}

```

Εικόνα 51 loginbtn\_Click web app

signupbtn\_Click: μεταφέρει το χρήστη στη σελίδα εγγραφής νέων χρηστών. (βλ. Εικόνα 52)

```

64 protected void signupbtn_Click(object sender, EventArgs e)
65 {
66     Response.Redirect("Signup.aspx");
67 }
68 }
69

```

Εικόνα 52 signup\_Click login web app

### 4.2.2. Sign up

Signup.aspx.cs

signupbtn\_Click: κάνει τους απαραίτητους ελέγχους και αν βρει ΑΜΚΑ ήδη καταχωρημένο στη βάση ίδιο με αυτό που πληκτρολόγησε ο χρήστης χωρίς να έχει λογαριασμό τότε ελέγχει απλά τα πεδία ΑΜΚΑ, Username, Password, ConfirmPassword. Αν δε βρει το ΑΜΚΑ στη βάση τότε κάνει τους απαραίτητους ελέγχους σε όλα τα πεδία και τέλος δημιουργεί λογαριασμό στο χρήστη. (βλ. Εικόνα 53)

```
protected void signuptbn_Click(object sender, EventArgs e)
{
    SqlConnection sqlCon = new SqlConnection(Con);
    if (sqlCon.State == ConnectionState.Closed)
    {
        sqlCon.Open();
    }
    SqlCommand comm = new SqlCommand("select * from Patients where AMKA='" + txtamka.Text + "'", sqlCon);
    if (txtamka.Text == "")
    {
        comm.Dispose();
        sqlCon.Close();
        Response.Write("<script>alert('Please enter something in AMKA box!');</script>");
    }
    else if (txtamka.Text != "")
    {
        if (userntxt.Text == "" || passtxt.Text == "" || conpasstxt.Text == "")
        {
            comm.Dispose();
            sqlCon.Close();
            Response.Write("<script>alert('Please enter something in every login credential box!');</script>");
        }
        else if (passtxt.Text != conpasstxt.Text)
        {
            comm.Dispose();
            sqlCon.Close();
            Response.Write("<script>alert('Password and confirm password are not the same!');</script>");
        }
        else
        {
            SqlDataReader rea = comm.ExecuteReader();
            if (rea.HasRows)
            {
                rea.Close();
                comm.Dispose();
                comm = new SqlCommand("select * from Patients where Username='" + userntxt.Text + "'", sqlCon);
                rea = comm.ExecuteReader();
                if (rea.HasRows)
                {
                    rea.Close();
                    comm.Dispose();
                    sqlCon.Close();
                    Response.Write("<script>alert('Username already exists!');</script>");
                }
            }
        }
    }
}
```

Εικόνα 53 signuptbn\_Click Sign up web app

### 4.2.3.Menu

Menu.aspx.cs

Page\_Load: όταν ανοίγει η σελίδα κάνει τους απαραίτητους ελέγχους για να δει αν ο χρήστης που προσπαθεί να αποκτήσει πρόσβαση στις λειτουργίες της είναι ο σωστός. Αν δεν είναι τον επαναφέρει πίσω στη σελίδα Login. **ΣΗΜΕΙΩΣΗ!!!** Ο συγκεκριμένος κώδικας χρησιμοποιείται σε όλες τις παρακάτω λειτουργίες που θα αναλυθούν στη συνέχεια. (βλ. Εικόνα 54)

```
13 protected void Page_Load(object sender, EventArgs e)
14 {
15     if (Session["userID"] == null || Session["userID"].Equals(""))
16     {
17         Response.Redirect("Login.aspx");
18     }
19     else if (Session["userID"].Equals("1"))
20     {
21         Label1.Text = "Hello " + Session["username"].ToString();
22     }
23     else
24     {
25         Response.Redirect("Login.aspx");
26     }
27 }
```

Εικόνα 54 loginbtn\_Click web app

LinkButton1\_Click (Logoutlabel): Πατώντας τη λέξη Logout ο χρήστης επαναφέρεται στην σελίδα Login και όλα τα sessionsχάνουν τις τιμές τους **ΣΗΜΕΙΩΣΗ!!!** Ο συγκεκριμένος κώδικας χρησιμοποιείται σε όλες τις παρακάτω λειτουργίες που θα αναλυθούν στη συνέχεια. (βλ. Εικόνα 55)

```
29 protected void LinkButton1_Click(object sender, EventArgs e)
30 {
31     Session["username"] = "";
32     Session["amka"] = "";
33     Session["userID"] = "";
34     Response.Redirect("Login.aspx");
35 }
```

Εικόνα 55 LinkButton1\_Click web app

#### 4.2.4. Patients

Patients.aspx.cs

savebtn\_Click: κάνει τη σύνδεση με τη βάση δεδομένων, έπειτα ελέγχει αν όλα τα πεδία έχουν συμπληρωθεί, και στη συνέχεια κάνει τους απαραίτητους ελέγχους για να δει αν θα αποθηκεύσει τα δεδομένα στους κατάλληλους πίνακες της βάσης ή αν θα δείξει άλλα σχετικά μηνύματα. (βλ. Εικόνα 56)

```

SqlConnection sqlCon = new SqlConnection(Con);
if (sqlCon.State == ConnectionState.Closed)
{
    sqlCon.Open();
}
SqlCommand comm = new SqlCommand("select * from Patients where AMKA='" + txtamka.Text + "'", sqlCon);
if (txtname.Text == "" || txtage.Text == "" || drpgender.Text == "" || txtphone.Text == "" || txtweight.Text == "" || txtheight.Text == "" || drpblood.Text == "")
{
    comm.Dispose();
    sqlCon.Close();
    Response.Write("<script>alert('Please enter something in every field!');</script>");
}
else
{
    SqlDataReader rea = comm.ExecuteReader();
    if (rea.HasRows)
    {
        rea.Close();
        comm.Dispose();
        sqlCon.Close();
        Response.Write("<script>alert('AMKA already exists!');</script>");
    }
    else
    {
        rea.Close();
        comm.Dispose();
        comm = new SqlCommand("Insert into Patients values(@AMKA,@Name,@Address,@Age,@Gender,@PhoneNumber,@Height,@Weight,@BloodType,@AddedDate,@UpdatedDate,@Username)", sqlCon);
        comm.Parameters.Add(new SqlParameter("@AMKA", txtamka.Text.Trim()));
        comm.Parameters.Add(new SqlParameter("@Name", txtname.Text.Trim()));
        comm.Parameters.Add(new SqlParameter("@Address", txtaddress.Text.Trim()));
        comm.Parameters.Add(new SqlParameter("@Age", Convert.ToInt64(txtage.Text)));
        comm.Parameters.Add(new SqlParameter("@Gender", drpgender.Text));
        comm.Parameters.Add(new SqlParameter("@PhoneNumber", txtphone.Text));
        comm.Parameters.Add(new SqlParameter("@Height", Convert.ToInt64(txtheight.Text)));
        comm.Parameters.Add(new SqlParameter("@Weight", Convert.ToInt64(txtweight.Text)));
        comm.Parameters.Add(new SqlParameter("@BloodType", drpblood.Text));
        comm.Parameters.Add(new SqlParameter("@AddedDate", DateTime.Now));
        comm.Parameters.Add(new SqlParameter("@UpdatedDate", DateTime.Now));
        comm.Parameters.Add(new SqlParameter("@Username", DBNull.Value));
        comm.ExecuteNonQuery();
        comm.Dispose();
        sqlCon.Close();
    }
}

```

Εικόνα 56 savebtn\_Click patients web app

updatebtn\_Click: κάνει τη σύνδεση με τη βάση δεδομένων, έπειτα ελέγχει αν όλα τα απαραίτητα πεδία έχουν συμπληρωθεί, και στη συνέχεια κάνει τους απαραίτητους ελέγχους για να δει αν θα αποθηκεύσει τα δεδομένα στους κατάλληλους πίνακες της βάσης ή αν θα δείξει άλλα σχετικά μηνύματα. (βλ. Εικόνα 57)

```

83 {
84     SqlConnection sqlCon = new SqlConnection(Con);
85     if (sqlCon.State == ConnectionState.Closed)
86     {
87         sqlCon.Open();
88     }
89     SqlCommand comm = new SqlCommand("select * from Patients where AMKA='" + txtamka.Text + "'", sqlCon);
90     if (txtamka.Text == "")
91     {
92         comm.Dispose();
93         sqlCon.Close();
94         Response.Write("<script>alert('Please enter something in the AMKA box!');</script>");
95     }
96     else if (txtname.Text == "" && txtage.Text == "" && drpgender.Text == "" && txtphone.Text == "" && txtweight.Text == "" && txtheight.Text == "" &&
97     {
98         comm.Dispose();
99         sqlCon.Close();
100        Response.Write("<script>alert('Please enter something in at least one field!');</script>");
101    }
102    else
103    {
104        SqlDataReader rea = comm.ExecuteReader();
105        if (rea.HasRows)
106        {
107            rea.Close();
108            if (txtname.Text != "")
109            {
110                comm.Dispose();
111                comm = new SqlCommand("Update Patients set Name=@Name where AMKA=@AMKA", sqlCon);
112                comm.Parameters.Add(new SqlParameter("@AMKA", txtamka.Text.Trim()));
113                comm.Parameters.Add(new SqlParameter("@Name", txtname.Text.Trim()));
114                comm.ExecuteNonQuery();
115                comm.Dispose();
116            }
117            if (txtaddress.Text != "")
118            {
119                comm.Dispose();
120                comm = new SqlCommand("Update Patients set Address=@Address where AMKA=@AMKA", sqlCon);
121                comm.Parameters.Add(new SqlParameter("@AMKA", txtamka.Text.Trim()));
122                comm.Parameters.Add(new SqlParameter("@Address", txtaddress.Text.Trim()));
123                comm.ExecuteNonQuery();
124                comm.Dispose();
125            }
126        }
127    }
128 }

```

Εικόνα 57 updatebtn\_Click patients web app

deletebtn\_Click: κάνει τη σύνδεση με τη βάση δεδομένων, έπειτα ελέγχει αν όλα τα απαραίτητα πεδία έχουν συμπληρωθεί, και στη συνέχεια κάνει τους απαραίτητους ελέγχους για να δει αν θα διαγράψει τα δεδομένα από όλους τους πίνακες της βάσης, οι οποίοι έχουν πληροφορίες σχετικές με το AMKA που θέλει να διαγράψει ο ιατρός, ή αν θα δείξει άλλα σχετικά μηνύματα. (βλ. Εικόνα 58)

```

558     {
559         SqlConnection sqlCon = new SqlConnection(Con);
560         if (sqlCon.State == ConnectionState.Closed)
561         {
562             sqlCon.Open();
563         }
564         SqlCommand comm = new SqlCommand("select * from Patients where AMKA='" + txtamka.Text + "'", sqlCon);
565         if (txtamka.Text == "")
566         {
567             comm.Dispose();
568             sqlCon.Close();
569             Response.Write("<script>alert('Please enter something in the AMKA box!');</script>");
570         }
571         else
572         {
573             SqlDataReader rea = comm.ExecuteReader();
574             if (rea.HasRows)
575             {
576                 rea.Close();
577                 comm.Dispose();
578                 comm = new SqlCommand("Delete from Patients where AMKA=@AMKA", sqlCon);
579                 comm.Parameters.Add(new SqlParameter("@AMKA", txtamka.Text.Trim()));
580                 comm.ExecuteNonQuery();
581                 comm.Dispose();
582                 comm = new SqlCommand("Delete from Conclusions where AMKA=@AMKA", sqlCon);
583                 comm.Parameters.Add(new SqlParameter("@AMKA", txtamka.Text.Trim()));
584                 comm.ExecuteNonQuery();
585                 comm.Dispose();
586                 comm = new SqlCommand("Delete from [User] where AMKA=@AMKA", sqlCon);
587                 comm.Parameters.Add(new SqlParameter("@AMKA", txtamka.Text.Trim()));
588                 comm.ExecuteNonQuery();
589                 comm.Dispose();
590                 comm = new SqlCommand("Delete from BloodInfo where AMKA=@AMKA", sqlCon);
591                 comm.Parameters.Add(new SqlParameter("@AMKA", txtamka.Text.Trim()));
592                 comm.ExecuteNonQuery();
593                 comm.Dispose();
594                 comm = new SqlCommand("Delete from Appointments where AMKA=@AMKA", sqlCon);
595                 comm.Parameters.Add(new SqlParameter("@AMKA", txtamka.Text.Trim()));
596                 comm.ExecuteNonQuery();
597                 sqlCon.Close();
598                 txtamka.Text = "";
599                 Response.Write("<script>alert('Το AMKA είναι δελεσμένο!');</script>");
600             }
601         }
602     }

```

Εικόνα 58 deletebtn\_Click patients web app

### 4.2.5.History

History.aspx.cs

Page\_Load: όταν ανοίγει η φόρμα εμφανίζει στο GridView1 τα δεδομένα που υπάρχουν στο συνδυασμό των πινάκων Patients-Conclusions της βάσης μας. (βλ. Εικόνα 59)

```

18     protected void Page_Load(object sender, EventArgs e)
19     {
20         if (Session["userID"] == null || Session["userID"].Equals(""))
21         {
22             Response.Redirect("Login.aspx");
23         }
24         else if (Session["userID"].Equals("1"))
25         {
26             Label1.Text = "Hello " + Session["username"].ToString();
27             SqlConnection sqlCon = new SqlConnection(Con);
28             SqlCommand comm = new SqlCommand("select * from Patients inner join Conclusions on Patients.AMKA = Conclusions.AMKA", sqlCon);
29             SqlDataAdapter sda = new SqlDataAdapter();
30             comm.Connection = sqlCon;
31             sqlCon.Open();
32             sda.SelectCommand = comm;
33             DataSet ds = new DataSet();
34             sda.Fill(ds);
35
36             GridView1.DataSource = ds.Tables[0];
37             GridView1.DataBind();
38             sqlCon.Close();
39         }
40         else
41         {
42             Response.Redirect("Login.aspx");
43         }
44     }

```

Εικόνα 59 Page\_Load history web app



txtamka\_TextChanged: κάνει αναζήτηση στο συνδυασμό πινάκων Patients-Conclusions των ΑΜΚΑ που μοιάζουν με τα δεδομένα που πληκτρολογεί ο χρήστης στο πεδίο αναζήτησης. (βλ. Εικόνα 60)

```
1 protected void txtamka_TextChanged(object sender, EventArgs e)
2 {
3     if (txtamka.Text != "")
4     {
5         int AMKA = Convert.ToInt32(txtamka.Text);
6         SqlConnection sqlCon = new SqlConnection(Con);
7         SqlCommand comm = new SqlCommand("select * from Patients, Conclusions where Patients.AMKA = Conclusions.AMKA and Patients.AMKA Like '" + AMKA + "%' a
8         SqlDataAdapter sda = new SqlDataAdapter(comm);
9         comm.Connection = sqlCon;
10        sqlCon.Open();
11        sda.SelectCommand = comm;
12        DataSet ds = new DataSet();
13        sda.Fill(ds);
14
15        GridView1.DataSource = ds.Tables[0];
16        GridView1.DataBind();
17        sqlCon.Close();
18    }
19    else
20    {
21        LoadData();
22    }
23 }
```

Εικόνα 60 txtamka\_TextChanged web app

LoadData: επαναφέρει τον πίνακα δεδομένων στην αρχική του κατάσταση αν το πεδίο αναζήτησης είναι κενό. (βλ. Εικόνα 61)

```
46 protected void LoadData()
47 {
48     SqlConnection sqlCon = new SqlConnection(Con);
49     SqlCommand comm = new SqlCommand("select * from Patients inner join Conclusions on Patients.AMKA = Conclusions.AMKA", sqlCon);
50     SqlDataAdapter sda = new SqlDataAdapter(comm);
51     comm.Connection = sqlCon;
52     sqlCon.Open();
53     sda.SelectCommand = comm;
54     DataSet ds = new DataSet();
55     sda.Fill(ds);
56
57     GridView1.DataSource = ds.Tables[0];
58     GridView1.DataBind();
59     sqlCon.Close();
60 }
```

Εικόνα 61 LoadData web app

#### 4.2.6. Blood stats

Bloodstats.aspx.cs

Page\_Load: όταν ανοίγει η φόρμα εμφανίζει στον πίνακα GridView1 τα δεδομένα που υπάρχουν στον πίνακα Blood Infoτης βάσης μας, ενώ εμφανίζει στο πίνακα GridView2 τα δεδομένα που υπάρχουν στο πίνακα Blood Averageτης βάσης μας. (βλ. Εικόνα 62)

```
17 protected void Page_Load(object sender, EventArgs e)
18 {
19     if (Session["userID"] == null || Session["userID"].Equals(""))
20     {
21         Response.Redirect("Login.aspx");
22     }
23     else if (Session["userID"].Equals("1"))
24     {
25         Label1.Text = "Hello " + Session["username"].ToString();
26         SqlConnection sqlCon = new SqlConnection(Con);
27         SqlCommand comm = new SqlCommand("select * from BloodInfo", sqlCon);
28         SqlDataAdapter sda = new SqlDataAdapter();
29         comm.Connection = sqlCon;
30         sqlCon.Open();
31         sda.SelectCommand = comm;
32         DataSet ds = new DataSet();
33         sda.Fill(ds);
34
35         GridView1.DataSource = ds.Tables[0];
36         GridView1.DataBind();
37         sqlCon.Close();
38
39         sqlCon = new SqlConnection(Con);
40         comm = new SqlCommand("select * from BloodAverage", sqlCon);
41         sda = new SqlDataAdapter();
42         comm.Connection = sqlCon;
43         sqlCon.Open();
44         sda.SelectCommand = comm;
45         ds = new DataSet();
46         sda.Fill(ds);
47
48         GridView2.DataSource = ds.Tables[0];
49         GridView2.DataBind();
50         sqlCon.Close();
51     }
52     else
53     {
54         Response.Redirect("Login.aspx");
55     }
56 }
```

Εικόνα 62 Page\_Load blood web app

savebtn\_Click: κάνει τη σύνδεση με τη βάση δεδομένων, έπειτα ελέγχει αν όλα τα πεδία έχουν συμπληρωθεί, και στη συνέχεια κάνει τους απαραίτητους ελέγχους για να δει αν θα αποθηκεύσει τα δεδομένα στους κατάλληλους πίνακες της βάσης ή αν θα δείξει άλλα σχετικά μηνύματα. (βλ. Εικόνα 63)



```

        SqlConnection sqlCon = new SqlConnection(Con);
        if (sqlCon.State == ConnectionState.Closed)
        {
            sqlCon.Open();
        }
        SqlCommand comm = new SqlCommand("select * from Patients where AMKA='" + txtamka.Text + "'", sqlCon);
        if (txtamka.Text == "" || txtabc.Text == "" || txtneut.Text == "" || txtlymph.Text == "" || txtmono.Text == "" || txteos.Text == "" || txtbaso.Text == ""
        {
            comm.Dispose();
            sqlCon.Close();
            Response.Write("<script>alert('Please enter something in every field!');</script>");
        }
        else
        {
            SqlDataReader rea = comm.ExecuteReader();
            if (rea.HasRows)
            {
                rea.Close();
                comm.Dispose();
                comm = new SqlCommand("select * from BloodInfo where AMKA='" + txtamka.Text + "'", sqlCon);
                rea = comm.ExecuteReader();
                if (rea.HasRows)
                {
                    rea.Close();
                    comm.Dispose();
                    sqlCon.Close();
                    Response.Write("<script>alert('AMKA already exists!');</script>");
                }
                else
                {
                    rea.Close();
                    comm.Dispose();
                    comm = new SqlCommand("Insert into BloodInfo values(@AMKA, @WBC, @NEUT, @LYMPH, @MONO, @EOS, @BASO, @RBC, @HGB, @HCT, @MCV, @MCH, @MCHC, @RDW, @PLT, @PCT, @HPT, @HPC, @HDF, @HDFI, @HDFI2, @HDFI3, @HDFI4, @HDFI5, @HDFI6, @HDFI7, @HDFI8, @HDFI9, @HDFI10, @HDFI11, @HDFI12, @HDFI13, @HDFI14, @HDFI15, @HDFI16, @HDFI17, @HDFI18, @HDFI19, @HDFI20, @HDFI21, @HDFI22, @HDFI23, @HDFI24, @HDFI25, @HDFI26, @HDFI27, @HDFI28, @HDFI29, @HDFI30, @HDFI31, @HDFI32, @HDFI33, @HDFI34, @HDFI35, @HDFI36, @HDFI37, @HDFI38, @HDFI39, @HDFI40, @HDFI41, @HDFI42, @HDFI43, @HDFI44, @HDFI45, @HDFI46, @HDFI47, @HDFI48, @HDFI49, @HDFI50, @HDFI51, @HDFI52, @HDFI53, @HDFI54, @HDFI55, @HDFI56, @HDFI57, @HDFI58, @HDFI59, @HDFI60, @HDFI61, @HDFI62, @HDFI63, @HDFI64, @HDFI65, @HDFI66, @HDFI67, @HDFI68, @HDFI69, @HDFI70, @HDFI71, @HDFI72, @HDFI73, @HDFI74, @HDFI75, @HDFI76, @HDFI77, @HDFI78, @HDFI79, @HDFI80, @HDFI81, @HDFI82, @HDFI83, @HDFI84, @HDFI85, @HDFI86, @HDFI87, @HDFI88, @HDFI89, @HDFI90, @HDFI91, @HDFI92, @HDFI93, @HDFI94, @HDFI95, @HDFI96, @HDFI97, @HDFI98, @HDFI99, @HDFI100)", sqlCon);
                    comm.Parameters.Add(new SqlParameter("@AMKA", txtamka.Text.Trim()));
                    comm.Parameters.Add(new SqlParameter("@WBC", Convert.ToDecimal(txtabc.Text)));
                    comm.Parameters.Add(new SqlParameter("@NEUT", Convert.ToDecimal(txtneut.Text)));
                    comm.Parameters.Add(new SqlParameter("@LYMPH", Convert.ToDecimal(txtlymph.Text)));
                    comm.Parameters.Add(new SqlParameter("@MONO", Convert.ToDecimal(txtmono.Text)));
                    comm.Parameters.Add(new SqlParameter("@EOS", Convert.ToDecimal(txteos.Text)));
                    comm.Parameters.Add(new SqlParameter("@BASO", Convert.ToDecimal(txtbaso.Text)));
                    comm.Parameters.Add(new SqlParameter("@RBC", Convert.ToDecimal(txttrbc.Text)));
                    comm.Parameters.Add(new SqlParameter("@HGB", Convert.ToDecimal(txttrhgb.Text)));
                    comm.Parameters.Add(new SqlParameter("@HCT", Convert.ToDecimal(txttrhct.Text)));
                    comm.Parameters.Add(new SqlParameter("@MCV", Convert.ToDecimal(txttrmcv.Text)));
                    comm.Parameters.Add(new SqlParameter("@MCH", Convert.ToDecimal(txttrmch.Text)));
                    comm.Parameters.Add(new SqlParameter("@MCHC", Convert.ToDecimal(txttrmchc.Text)));
                    comm.Parameters.Add(new SqlParameter("@RDW", Convert.ToDecimal(txttrrdw.Text)));
                    comm.Parameters.Add(new SqlParameter("@PLT", Convert.ToDecimal(txttrplt.Text)));
                    comm.Parameters.Add(new SqlParameter("@PCT", Convert.ToDecimal(txttrpct.Text)));
                    comm.Parameters.Add(new SqlParameter("@HPT", Convert.ToDecimal(txttrhpt.Text)));
                    comm.Parameters.Add(new SqlParameter("@HPC", Convert.ToDecimal(txttrhpc.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI", Convert.ToDecimal(txttrhdfi.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI2", Convert.ToDecimal(txttrhdfi2.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI3", Convert.ToDecimal(txttrhdfi3.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI4", Convert.ToDecimal(txttrhdfi4.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI5", Convert.ToDecimal(txttrhdfi5.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI6", Convert.ToDecimal(txttrhdfi6.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI7", Convert.ToDecimal(txttrhdfi7.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI8", Convert.ToDecimal(txttrhdfi8.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI9", Convert.ToDecimal(txttrhdfi9.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI10", Convert.ToDecimal(txttrhdfi10.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI11", Convert.ToDecimal(txttrhdfi11.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI12", Convert.ToDecimal(txttrhdfi12.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI13", Convert.ToDecimal(txttrhdfi13.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI14", Convert.ToDecimal(txttrhdfi14.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI15", Convert.ToDecimal(txttrhdfi15.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI16", Convert.ToDecimal(txttrhdfi16.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI17", Convert.ToDecimal(txttrhdfi17.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI18", Convert.ToDecimal(txttrhdfi18.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI19", Convert.ToDecimal(txttrhdfi19.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI20", Convert.ToDecimal(txttrhdfi20.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI21", Convert.ToDecimal(txttrhdfi21.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI22", Convert.ToDecimal(txttrhdfi22.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI23", Convert.ToDecimal(txttrhdfi23.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI24", Convert.ToDecimal(txttrhdfi24.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI25", Convert.ToDecimal(txttrhdfi25.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI26", Convert.ToDecimal(txttrhdfi26.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI27", Convert.ToDecimal(txttrhdfi27.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI28", Convert.ToDecimal(txttrhdfi28.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI29", Convert.ToDecimal(txttrhdfi29.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI30", Convert.ToDecimal(txttrhdfi30.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI31", Convert.ToDecimal(txttrhdfi31.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI32", Convert.ToDecimal(txttrhdfi32.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI33", Convert.ToDecimal(txttrhdfi33.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI34", Convert.ToDecimal(txttrhdfi34.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI35", Convert.ToDecimal(txttrhdfi35.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI36", Convert.ToDecimal(txttrhdfi36.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI37", Convert.ToDecimal(txttrhdfi37.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI38", Convert.ToDecimal(txttrhdfi38.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI39", Convert.ToDecimal(txttrhdfi39.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI40", Convert.ToDecimal(txttrhdfi40.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI41", Convert.ToDecimal(txttrhdfi41.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI42", Convert.ToDecimal(txttrhdfi42.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI43", Convert.ToDecimal(txttrhdfi43.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI44", Convert.ToDecimal(txttrhdfi44.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI45", Convert.ToDecimal(txttrhdfi45.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI46", Convert.ToDecimal(txttrhdfi46.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI47", Convert.ToDecimal(txttrhdfi47.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI48", Convert.ToDecimal(txttrhdfi48.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI49", Convert.ToDecimal(txttrhdfi49.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI50", Convert.ToDecimal(txttrhdfi50.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI51", Convert.ToDecimal(txttrhdfi51.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI52", Convert.ToDecimal(txttrhdfi52.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI53", Convert.ToDecimal(txttrhdfi53.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI54", Convert.ToDecimal(txttrhdfi54.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI55", Convert.ToDecimal(txttrhdfi55.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI56", Convert.ToDecimal(txttrhdfi56.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI57", Convert.ToDecimal(txttrhdfi57.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI58", Convert.ToDecimal(txttrhdfi58.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI59", Convert.ToDecimal(txttrhdfi59.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI60", Convert.ToDecimal(txttrhdfi60.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI61", Convert.ToDecimal(txttrhdfi61.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI62", Convert.ToDecimal(txttrhdfi62.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI63", Convert.ToDecimal(txttrhdfi63.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI64", Convert.ToDecimal(txttrhdfi64.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI65", Convert.ToDecimal(txttrhdfi65.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI66", Convert.ToDecimal(txttrhdfi66.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI67", Convert.ToDecimal(txttrhdfi67.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI68", Convert.ToDecimal(txttrhdfi68.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI69", Convert.ToDecimal(txttrhdfi69.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI70", Convert.ToDecimal(txttrhdfi70.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI71", Convert.ToDecimal(txttrhdfi71.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI72", Convert.ToDecimal(txttrhdfi72.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI73", Convert.ToDecimal(txttrhdfi73.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI74", Convert.ToDecimal(txttrhdfi74.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI75", Convert.ToDecimal(txttrhdfi75.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI76", Convert.ToDecimal(txttrhdfi76.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI77", Convert.ToDecimal(txttrhdfi77.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI78", Convert.ToDecimal(txttrhdfi78.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI79", Convert.ToDecimal(txttrhdfi79.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI80", Convert.ToDecimal(txttrhdfi80.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI81", Convert.ToDecimal(txttrhdfi81.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI82", Convert.ToDecimal(txttrhdfi82.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI83", Convert.ToDecimal(txttrhdfi83.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI84", Convert.ToDecimal(txttrhdfi84.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI85", Convert.ToDecimal(txttrhdfi85.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI86", Convert.ToDecimal(txttrhdfi86.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI87", Convert.ToDecimal(txttrhdfi87.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI88", Convert.ToDecimal(txttrhdfi88.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI89", Convert.ToDecimal(txttrhdfi89.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI90", Convert.ToDecimal(txttrhdfi90.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI91", Convert.ToDecimal(txttrhdfi91.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI92", Convert.ToDecimal(txttrhdfi92.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI93", Convert.ToDecimal(txttrhdfi93.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI94", Convert.ToDecimal(txttrhdfi94.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI95", Convert.ToDecimal(txttrhdfi95.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI96", Convert.ToDecimal(txttrhdfi96.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI97", Convert.ToDecimal(txttrhdfi97.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI98", Convert.ToDecimal(txttrhdfi98.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI99", Convert.ToDecimal(txttrhdfi99.Text)));
                    comm.Parameters.Add(new SqlParameter("@HDFI100", Convert.ToDecimal(txttrhdfi100.Text)));
                    comm.ExecuteNonQuery();
                }
            }
        }
    }
}

```

Εικόνα 63 savebtn\_Click blood web app

updatebtn\_Click: κάνει τη σύνδεση με τη βάση δεδομένων, έπειτα ελέγχει αν όλα τα απαραίτητα πεδία έχουν συμπληρωθεί, και στη συνέχεια κάνει τους απαραίτητους ελέγχους για να δει αν θα αποθηκεύσει τα δεδομένα στους κατάλληλους πίνακες της βάσης ή αν θα δείξει άλλα σχετικά μηνύματα. (βλ. Εικόνα 64)

```

69 protected void updatebtn_Click(object sender, EventArgs e)
70 {
71     SqlConnection sqlCon = new SqlConnection(Con);
72     if (sqlCon.State == ConnectionState.Closed)
73     {
74         sqlCon.Open();
75     }
76     SqlCommand comm = new SqlCommand("select * from BloodInfo where AMKA='" + txtamka.Text + "'", sqlCon);
77     if (txtamka.Text == "")
78     {
79         comm.Dispose();
80         sqlCon.Close();
81         Response.Write("<script>alert('Please enter something in the AMKA box!');</script>");
82     }
83     else if (txtamka.Text == "" && txtwbc.Text == "" && txtneut.Text == "" && txtlymph.Text == "" && txtmono.Text == "" && txteos.Text == "" &
84     {
85         comm.Dispose();
86         sqlCon.Close();
87         Response.Write("<script>alert('Please enter something in at least one field!');</script>");
88     }
89     else
90     {
91         SqlDataReader rea = comm.ExecuteReader();
92         if (rea.HasRows)
93         {
94             rea.Close();
95             if (txtwbc.Text != "")
96             {
97                 comm.CommandText = "Update BloodInfo set WBC=@WBC where AMKA=@AMKA";
98                 comm.Parameters.Add(new SqlParameter("@AMKA", txtamka.Text.Trim()));
99                 comm.Parameters.Add(new SqlParameter("@WBC", Convert.ToDecimal(txtwbc.Text)));
100                comm.ExecuteNonQuery();
101                comm.Dispose();
102            }
103            if (txtneut.Text != "")
104            {
105                comm.CommandText = "Update BloodInfo set NEUT=@NEUT where AMKA=@AMKA";
106                comm.Parameters.Add(new SqlParameter("@AMKA", txtamka.Text.Trim()));
107                comm.Parameters.Add(new SqlParameter("@NEUT", Convert.ToDecimal(txtneut.Text)));
108                comm.ExecuteNonQuery();
109                comm.Dispose();
110            }
111            if (txtlymph.Text != "")

```

Εικόνα 64 updatebtn\_Click blood web app

deletebtn\_Click: κάνει τη σύνδεση με τη βάση δεδομένων, έπειτα ελέγχει αν όλα τα απαραίτητα πεδία έχουν συμπληρωθεί, και στη συνέχεια κάνει τους απαραίτητους ελέγχους για να δει αν θα διαγράψει τα δεδομένα μόνο από το πίνακα BloodInfo της βάσης. (βλ.

Εικόνα 65)

```

98 protected void deletebtn_Click(object sender, EventArgs e)
99 {
100     SqlConnection sqlCon = new SqlConnection(Con);
101     if (sqlCon.State == ConnectionState.Closed)
102     {
103         sqlCon.Open();
104     }
105     SqlCommand comm = new SqlCommand("select * from BloodInfo where AMKA='" + txtamka.Text + "'", sqlCon);
106     if (txtamka.Text == "")
107     {
108         comm.Dispose();
109         sqlCon.Close();
110         Response.Write("<script>alert('Please enter something in the AMKA box!');</script>");
111     }
112     else
113     {
114         SqlDataReader rea = comm.ExecuteReader();
115         if (rea.HasRows)
116         {
117             rea.Close();
118             comm.Dispose();
119             comm = new SqlCommand("Delete from BloodInfo where AMKA=@AMKA", sqlCon);
120             comm.Parameters.Add(new SqlParameter("@AMKA", txtamka.Text.Trim()));
121             comm.ExecuteNonQuery();
122             comm.Dispose();
123             sqlCon.Close();
124             txtamka.Text = "";
125             Response.Write("<script>alert('Information deleted!');</script>");
126
127             GridView1.DataBind();
128         }
129         else
130         {
131             rea.Close();
132             comm.Dispose();
133             sqlCon.Close();
134             txtamka.Text = "";
135             Response.Write("<script>alert('AMKA doesnt exist in database!');</script>");
136         }
137     }
138 }

```

Εικόνα 65 deletebtn\_Click blood web app

txtsrch\_TextChanged: κάνει αναζήτηση στο πίνακα BloodInfo των AMKA που μοιάζουν με τα δεδομένα που πληκτρολογεί ο χρήστης στο πεδίο αναζήτησης. (βλ. Εικόνα 66)

```

98 protected void txtsrch_TextChanged(object sender, EventArgs e)
99 {
100     if (txtsrch.Text != "")
101     {
102         int AMKA = Convert.ToInt32(txtsrch.Text);
103         SqlConnection sqlCon = new SqlConnection(Con);
104         SqlCommand comm = new SqlCommand("select * from BloodInfo where AMKA like '" + AMKA + "%'", sqlCon);
105         SqlDataAdapter sda = new SqlDataAdapter(comm);
106         comm.Connection = sqlCon;
107         sqlCon.Open();
108         sda.SelectCommand = comm;
109         DataSet ds = new DataSet();
110         sda.Fill(ds);
111
112         GridView1.DataSource = ds.Tables[0];
113         GridView1.DataBind();
114         sqlCon.Close();
115     }
116     else
117     {
118         LoadData();
119     }
120 }

```

Εικόνα 66 txtsrch\_TextChanged web app

### 4.2.7.Appointments

Appointments.aspx.cs

Page\_Load: με το άνοιγμα της σελίδας ελέγχει αν το session είναι σωστό και έπειτα ψάχνει για να βρει αν ο χρήστης που έκανε σύνδεση έχει ήδη κάποιο εκκρεμή ραντεβού με το ιατρό. Αν έχει τότε εμφανίζεται το κουμπί deletebtn, αλλιώς μένει αόρατο μέχρι ο χρήστης να κλείσει κάποιο ραντεβού. (βλ. Εικόνα 67)



```
1 {
2     //If (!IsPostBack)
3     //{
4         Calendar1.SelectedDate = DateTime.Now.Date;
5     //}
6     if (Session["userID"] == null || Session["userID"].Equals(""))
7     {
8         Response.Redirect("Login.aspx");
9     }
10    else if (Session["userID"].Equals("3"))
11    {
12        Label1.Text = "Hello " + Session["username"].ToString();
13        SqlConnection sqlCon = new SqlConnection(Con);
14        if (sqlCon.State == ConnectionState.Closed)
15        {
16            sqlCon.Open();
17        }
18        SqlCommand comm = new SqlCommand("select * from Appointments where AMKA='"+ Session["amka"].ToString() + "' and AppointmentDate >='"+
19        SqlDataReader rea = comm.ExecuteReader();
20        if (rea.HasRows)
21        {
22            rea.Close();
23            comm.Dispose();
24            sqlCon.Close();
25            deletebtn.Visible = true;
26        }
27        else
28        {
29            rea.Close();
30            comm.Dispose();
31            sqlCon.Close();
32            deletebtn.Visible = false;
33        }
34    }
35    else
36    {
37        Response.Redirect("Login.aspx");
38    }
39 }
```

Εικόνα 67 Page\_Load appointments web app

check\_Click: ελέγχει αν υπάρχει διαθέσιμο ραντεβού για την ώρα που επέλεξε ο χρήστης και βγάζει σχετικό μήνυμα. (βλ. Εικόνα 68)

```

35 {
36     choice = Calendar1.SelectedDate.Year + "-" + Calendar1.SelectedDate.Month + "-" + Calendar1.SelectedDate.Day + " " + DropDownList1.Text;
37     //Response.Write(choice);
38     SqlConnection sqlCon = new SqlConnection(Con);
39     if (sqlCon.State == ConnectionState.Closed)
40     {
41         sqlCon.Open();
42     }
43     SqlCommand comm = new SqlCommand("select * from Appointments where AppointmentDate='" + choice + "'", sqlCon);
44     SqlDataReader rea = comm.ExecuteReader();
45     if (rea.HasRows)
46     {
47         rea.Close();
48         comm.Dispose();
49         sqlCon.Close();
50         Response.Write("<script>alert('Sorry! There is no available appointment at the selected hour!');</script>");
51     }
52     else
53     {
54         if (DateTime.Parse(choice) < DateTime.Now)
55         {
56             rea.Close();
57             comm.Dispose();
58             sqlCon.Close();
59             Response.Write("<script>alert('Sorry! There is no available appointment at the selected hour!');</script>");
60         }
61         else
62         {
63             rea.Close();
64             comm.Dispose();
65             sqlCon.Close();
66             Response.Write("<script>alert('There is an appointment available for this hour!');</script>");
67         }
68     }
69 }

```

Εικόνα 68 check\_Click web app

book\_Click: ελέγχει αν υπάρχει διαθέσιμο ραντεβού για την ώρα που επέλεξε ο χρήστης και αν υπάρχει τότε το αποθηκεύει στη βάση μαζί με το ΑΜΚΑ του χρήστη και την ημερομηνία-ώρα που επέλεξε βγάζοντας έπειτα σχετικό μήνυμα. (βλ. Εικόνα 69)

```

32 {
33     choice = Calendar1.SelectedDate.Year + "-" + Calendar1.SelectedDate.Month + "-" + Calendar1.SelectedDate.Day + " " + DropDownList1.Text;
34     //Response.Write(choice);
35     SqlConnection sqlCon = new SqlConnection(Con);
36     if (sqlCon.State == ConnectionState.Closed)
37     {
38         sqlCon.Open();
39     }
40     SqlCommand comm = new SqlCommand("select * from Appointments where AppointmentDate='" + choice + "'", sqlCon);
41     SqlDataReader rea = comm.ExecuteReader();
42     if (rea.HasRows)
43     {
44         rea.Close();
45         comm.Dispose();
46         sqlCon.Close();
47         Response.Write("<script>alert('Sorry! There is no available appointment at the selected hour!');</script>");
48     }
49     else
50     {
51         rea.Close();
52         comm = new SqlCommand("select * from Appointments where ΑΜΚΑ='" + Session["amka"].ToString() + "' and AppointmentDate >='" + DateTime.Now + "'", sqlCon);
53         rea = comm.ExecuteReader();
54         if (rea.HasRows)
55         {
56             rea.Close();
57             comm.Dispose();
58             sqlCon.Close();
59             Response.Write("<script>alert('You have already set an appointment with the doctor! If you wish to set a new one, delete your current one and choose another hour!');</script>");
60         }
61         else
62         {
63             if (DateTime.Parse(choice) < DateTime.Now)
64             {
65                 rea.Close();
66                 comm.Dispose();
67                 sqlCon.Close();
68                 Response.Write("<script>alert('Sorry! There is no available appointment at the selected hour!');</script>");
69             }
70             else
71             {
72                 rea.Close();
73                 comm.Dispose();
74                 sqlCon.Close();
75                 Response.Write("<script>alert('Appointment successfully set!');</script>");
76             }
77         }
78     }
79 }

```

Εικόνα 69 book\_Click web app

deletebtn\_Click: διαγράφει το εκκρεμή ραντεβού που έχει ο χρήστης από τη βάση. (βλ. Εικόνα 70)

```

17 protected void deletebtn_Click(object sender, EventArgs e)
18 {
19     SqlConnection sqlCon = new SqlConnection(Con);
20     if (sqlCon.State == ConnectionState.Closed)
21     {
22         sqlCon.Open();
23     }
24     SqlCommand comm = new SqlCommand("select * from Appointments where AMKA='" + Session["amka"].ToString() + "' and AppointmentDate >='" + DateTime.Now + "'");
25     SqlDataReader rea = comm.ExecuteReader();
26     if (rea.HasRows)
27     {
28         rea.Close();
29         comm.Dispose();
30         comm = new SqlCommand("delete from Appointments where AMKA='" + Session["amka"].ToString() + "' and AppointmentDate >='" + DateTime.Now + "'");
31         comm.ExecuteNonQuery();
32         comm.Dispose();
33         sqlCon.Close();
34         Response.Write("<script>alert('Appointment deleted! You can now make a new one!');</script>");
35     }
36     else
37     {
38         rea.Close();
39         comm.Dispose();
40         sqlCon.Close();
41         Response.Write("<script>alert('You dont have a pending appointment at this moment!');</script>");
42     }
43 }

```

Εικόνα 70 deletebtn\_Click appointments web app

## 5.ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Το πρόγραμμα παρέχει ήδη βασικές λειτουργίες για την αποθήκευση δεδομένων κάνοντας κατάλληλους ελέγχους. Θα μπορούσε λοιπόν μελλοντικά να τροποποιηθεί εύκολα για να αποθηκεύει ιατρικά δεδομένα διαφόρων ειδικοτήτων (ουρολόγων, ογκολόγων, καρδιολόγων, κοκ) καθώς και να υποστηρίξει τη λειτουργία περισσότερων από 1 ειδικότητα ιατρικών θεμάτων. Επιπλέον, μια επιπλέον προσθήκη χρήσιμη στην εφαρμογή θα ήταν η πρόσβαση σε δεδομένα των ασθενών μέσω εφαρμογών όπως ΗΔΙΚΑ (Ηλεκτρονική Διακυβέρνηση Κοινωνικής Ασφάλισης), ΑΗΦΥ (Ατομικός Ηλεκτρονικός Φάκελος Υγείας) που υποστηρίζονται από τη διαδικτυακή σελίδα του κράτους (gov.gr) και έτσι να είναι εφικτή και η λειτουργία της άυλης συνταγογράφησης.

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

<https://www.athenahealth.com/>

<http://eclipsepracticemanagementsoftware.com/>

<https://myhealth.gov.gr/>

<https://www.epic.com/software>

<https://ehr.meditech.com/ehr-solutions>

<https://stackoverflow.com/>

<https://www.codeproject.com/>

<https://www.w3schools.com/>

<https://learn.microsoft.com/en-us/dotnet/>

<https://social.msdn.microsoft.com/Forums/en-US/home>