ITUCSDB1931 Documentation

ITUCSDB1931

Dec 27, 2019

CONTENTS

Team itucsdb1931

Members

- Doğu Ozan Kumru
- Talha Çomak

Project name is Pre-Diagnosis. The aim is provide better communication between doctors and patients. Thanks to Pre-Diagnosis, misdiagnosis resulting from incomplete information are avoided. Doctor does not waste time about patient information, so the inspection time will be shorten. Doctor know about patient's disease history and hereditary diseases,thus doctor can do better predictable about disease. This system is designed for doctors to see and update detailed informations about patients. Doctors can register the system and can login as user. Users can add new patients to the system. Users can also update informations of patients, but can not delete any user. Only admins can delete patients. Admins can also delete doctors from the system.

CHAPTER

ONE

USER GUIDE

User can register with enter required information in register page [1].

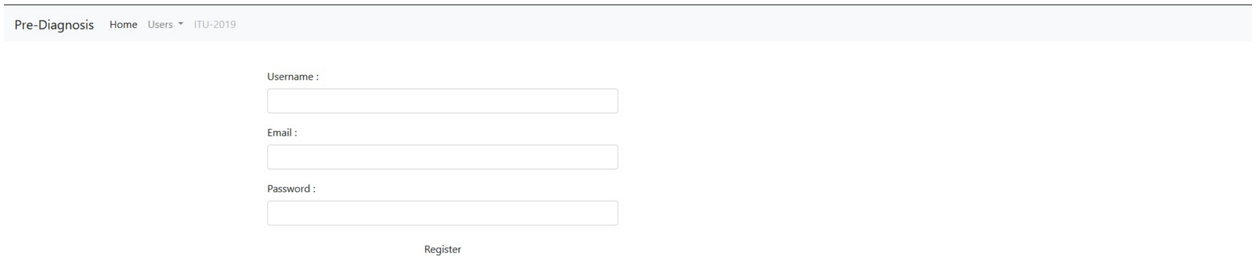


Fig. 1: Register page [1].

Then with this information user can login in login page [2].

User can check patient that desired to see information with box in the doctor page with entered the patient's ID. If there is new patient , user can add the patient to database with add patient link [3].

User can edit patient information. For edit operation user checks patient id that desired then use buttons near to subject that want to edit [4].

User can change old data with new one by update button. When user click update button box(s) appears and new data entered to box(s) [5].

User can delete patient's data. User clicks delete button and select data that will be deleted [6].

User can add multiple data to one subject with add new button [7].

Finally, user can access home, login and register pages with links on the bar that top of the screen easily [8].

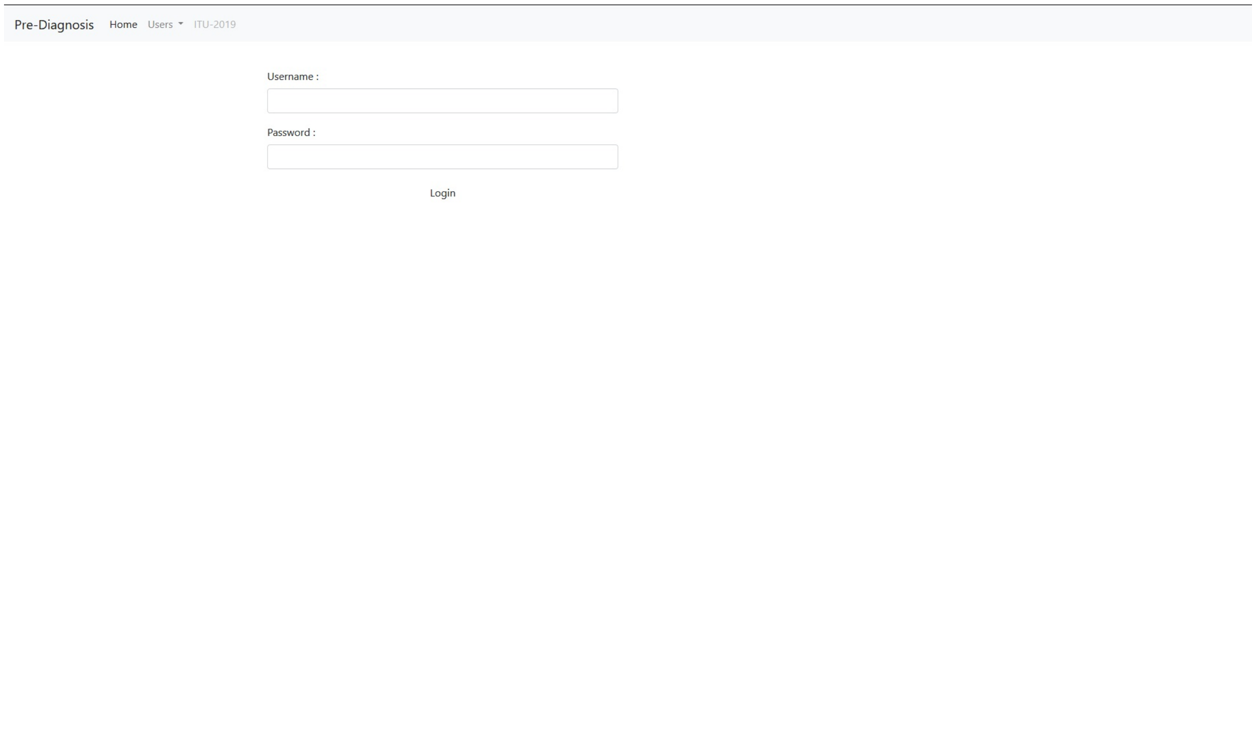


Fig. 2: Login page [2].

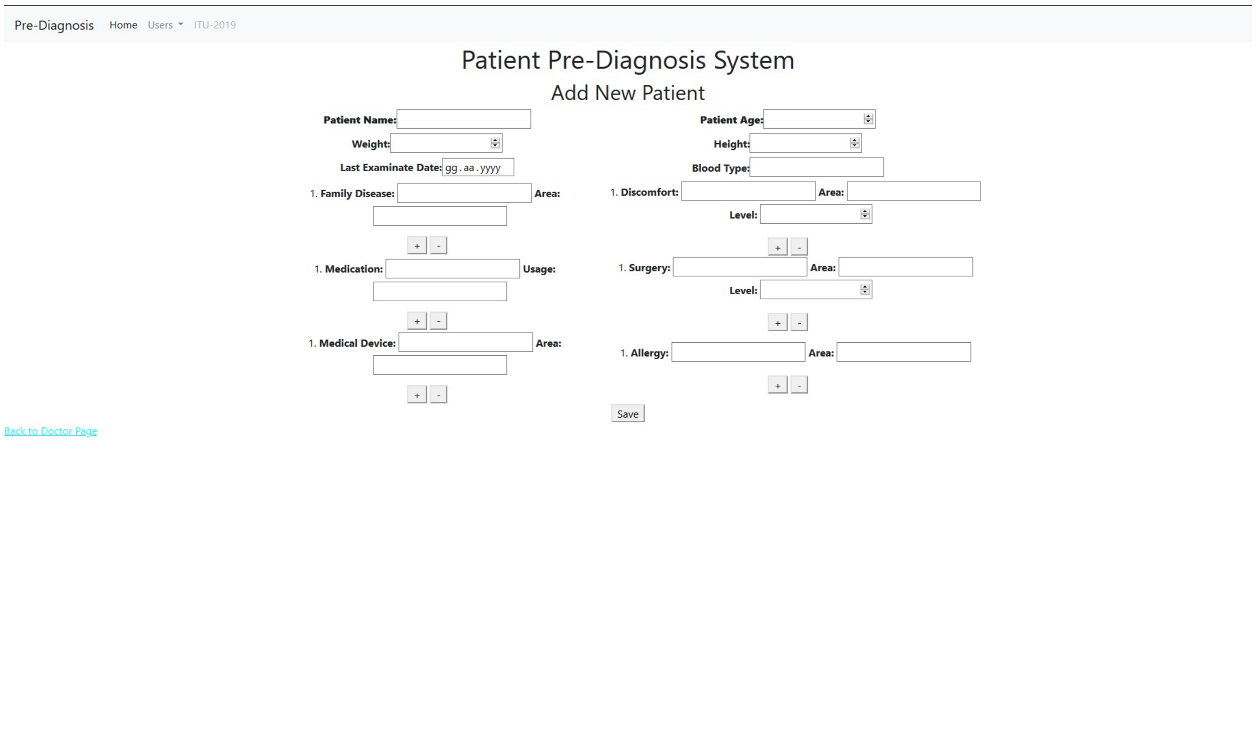
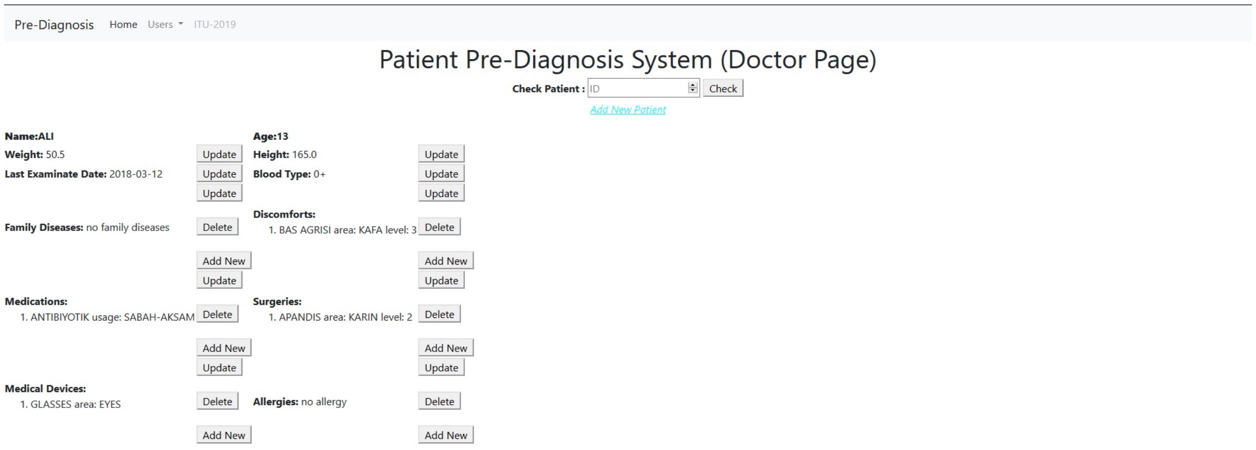


Fig. 3: Add new patient on Doctor page [3].



Patient Pre-Diagnosis System (Doctor Page)

Check Patient : ID

Name:ALI
Weight: 50.5
Last Examinate Date: 2018-03-12
Family Diseases: no family diseases
Medications:
1. ANTIBIYOTIK usage: SABAH-AKSAM

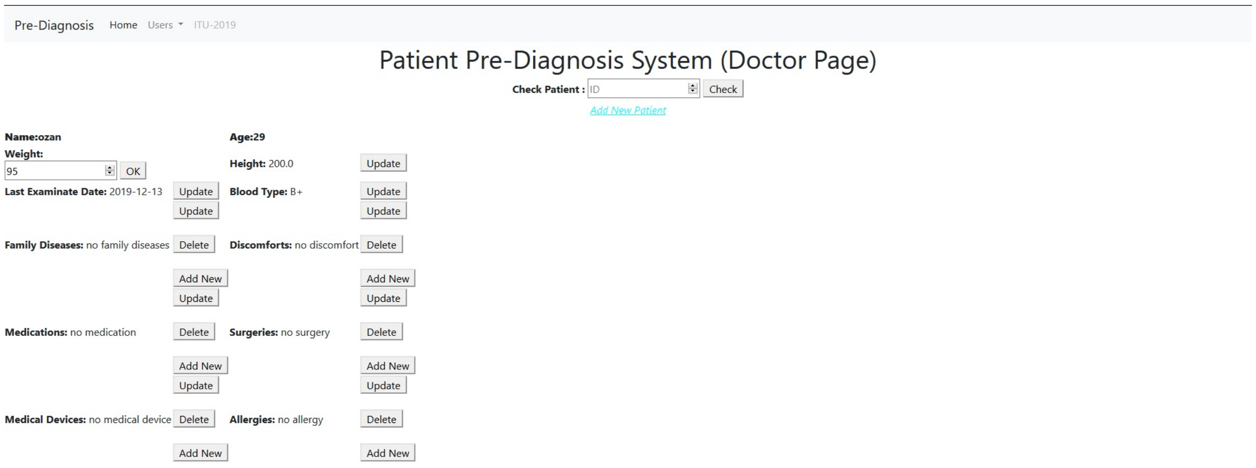
Medical Devices:
1. GLASSES area: EYES

Age:13
Height: 165.0
Blood Type: 0+
Discomforts:
1. BAS AGRISI area: KAFA level: 3

Surgeries:
1. APANDIS area: KARIN level: 2

Allergies: no allergy

Fig. 4: CRUD operations on patient [4].



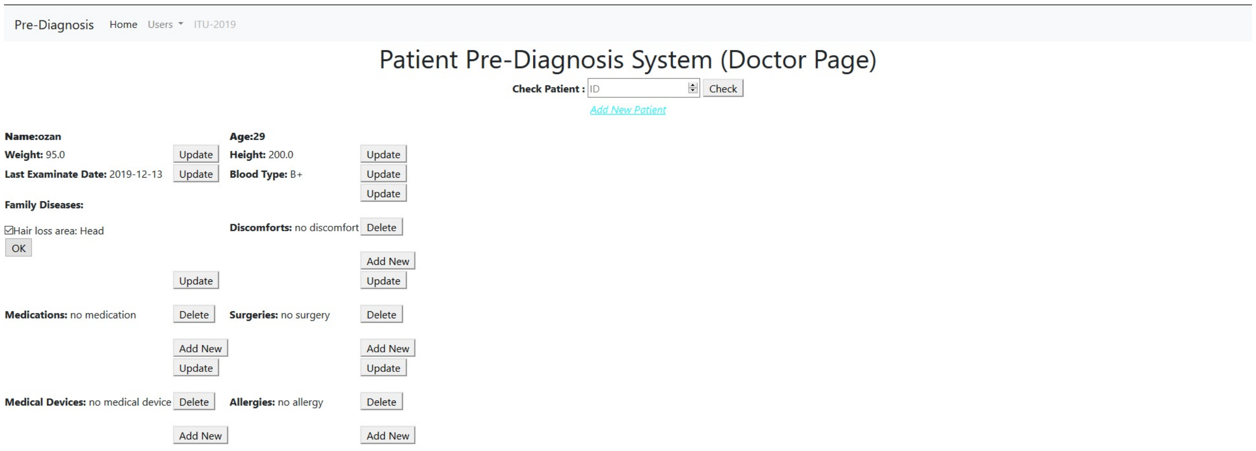
Patient Pre-Diagnosis System (Doctor Page)

Check Patient : ID

Name:ozan
Weight: 95
Last Examinate Date: 2019-12-13
Family Diseases: no family diseases
Medications: no medication
Medical Devices: no medical device

Age:29
Height: 200.0
Blood Type: B+
Discomforts: no discomfort
Surgeries: no surgery
Allergies: no allergy

Fig. 5: Update operations on patient [5].

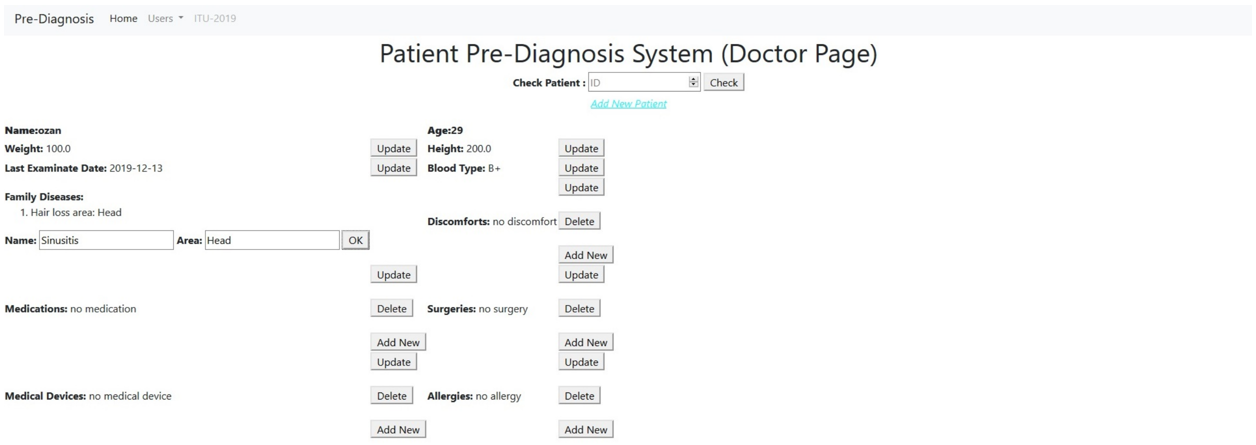


Patient Pre-Diagnosis System (Doctor Page)

Check Patient : ID

Name:ozan
Age:29
Weight: 95.0
Height: 200.0
Last Examinate Date: 2019-12-13
Blood Type: B+
Family Diseases:
☑Hair loss area: Head
Discomforts: no discomfort
Medications: no medication
Surgeries: no surgery
Medical Devices: no medical device
Allergies: no allergy

Fig. 6: Delete operations on patient [6].



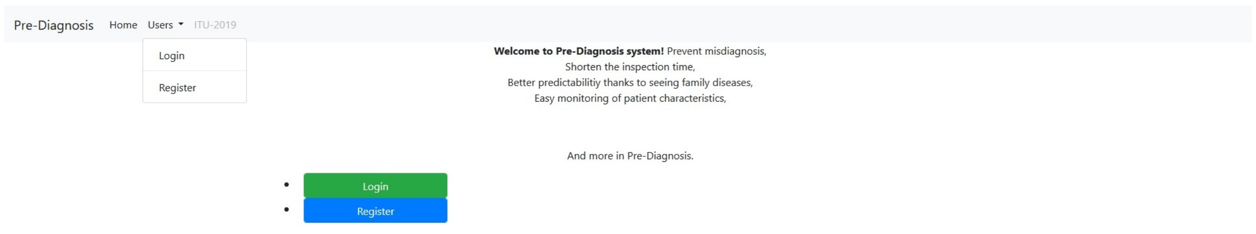
Patient Pre-Diagnosis System (Doctor Page)

Check Patient : ID

Name:ozan
Age:29
Weight: 100.0
Height: 200.0
Last Examinate Date: 2019-12-13
Blood Type: B+
Family Diseases:
1. Hair loss area: Head

Name: Sinusitis **Area:** Head
Discomforts: no discomfort
Medications: no medication
Surgeries: no surgery
Medical Devices: no medical device
Allergies: no allergy

Fig. 7: Add operations on patient [7].



Welcome to Pre-Diagnosis system! Prevent misdiagnosis,
Shorten the inspection time,
Better predictabilitiy thanks to seeing family diseases,
Easy monitoring of patient characteristics,

And more in Pre-Diagnosis.

Fig. 8: Navigation Bar [8].

1.1 Home, Login, Register and Admin Pages are Implemented by Doğu Ozan Kumru

1.2 Doctor page and Add new patient pages are Implemented by Talha Çomak

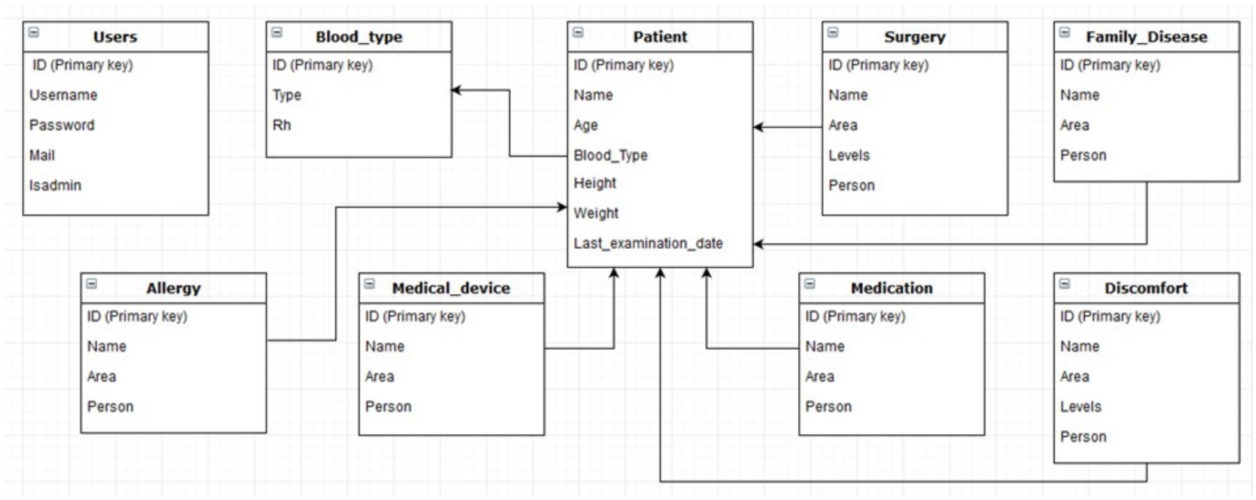
CHAPTER

TWO

DEVELOPER GUIDE

2.1 Database Design

Our website background consist of mainly Patient, Family_disease, Discomfort, Medical_device, Medication, Allergy and Surgery tables in the database. Addictivly User and Blood_type tables are exist. Patient table is the main table to store the basic informations about patients. The other tables except User are used for detailed informations about the patients and include a foreign key of the patient table. User table is used for keep login informations of doctors and admins.



2.2 Code

We done most of the operations in the server.py. We used @app.route methods to implement operations on back-end. All back-end operations are done in server.py. Front-end operations are implemented in server.py and the html files via jinja. Also we used ajax in the add_patient.html file to implement some front-end operations.

2.3 Server.py:

In the server.py we used such operations below. To start application:

```
#server.py

@app.route("/")
def home():
        return render_template('home.html')
```

To login the system:

```
#server.py

@app.route("/login", methods=["GET", "POST"])
def login():
        if request.method == "POST":
                uname = request.form["uname"]
                passw = request.form["passw"]

                state = "SELECT ID, ISADMIN, PASSWORD FROM USERS WHERE USERNAME='{}'".
↪format(uname)
                with dbapi2.connect(db_url) as connection:
                        cursor = connection.cursor()
                        cursor.execute(state)
                        record = cursor.fetchone()
                        if record != None:
                                if record[1]: # admin
                                        if hasher.verify(passw, record[2]):
                                                return redirect(url_for("admin_page"))
                                        else: ######################################␣
↪hatalı şifre
                                                render_template("login.html")
                                else: # doctor
                                        if hasher.verify(passw, record[2]):
                                                return render_template('doctor.html',␣
↪display="none")
                                        else:  ######################################␣
↪hatalı şifre
                                                render_template("login.html")
        return render_template("login.html")
```

To register the system:

```
#server.py

@app.route("/register", methods=["GET", "POST"])
def register():
        if request.method == "POST":
```

(continues on next page)

(continued from previous page)

```
                        uname = request.form['uname']
                        state = "SELECT ID FROM USERS WHERE USERNAME='{}'".format(uname)
                        with dbapi2.connect(db_url) as connection:
                                cursor = connection.cursor()
                                cursor.execute(state)
                                record = cursor.fetchone()
                                cursor.close()
                        if record == None:
                                mail = request.form['mail']
                                passw = request.form['passw']
                                hashed = hasher.hash(passw)
                                state = "INSERT INTO USERS(USERNAME, PASSWORD, MAIL) VALUES('
↪{}','{}','{}') ".format(uname, hashed, mail)
                                with dbapi2.connect(db_url) as connection:
                                        cursor = connection.cursor()
                                        cursor.execute(state)
                                        cursor.close()
                        else:
                                return render_template("register.html")
                        return redirect(url_for("login"))
                return render_template("register.html")
```

If admin is entered the system this function directing him/her the admin page:

```
#server.py

@app.route("/admin", methods=["GET", "POST"])
def admin_page():
        return render_template('admin.html', display="none")
```

If a user entered the system this function directing him/her the doctor page:

```
#server.py

@app.route("/doctor", methods=["GET", "POST"])
def doctor_page():
        return render_template('doctor.html', display="none")
```

Doctors who clicks add new patient button are directed the add patient page via this function:

```
#server.py

@app.route("/add_patient")
def add_patient():
        return render_template('add_patient.html')
```

Admins can see all patients on the admin page via this function:

```
#server.py
@app.route("/ShowAllPatients/",  methods=['GET', 'POST'])
def show_patients():
        state = "SELECT ALL * FROM PATIENT"
        with dbapi2.connect(db_url) as connection:
                cursor = connection.cursor()
                cursor.execute(state)
                patients = cursor.fetchall()
```

(continues on next page)

(continued from previous page)

```
                cursor.close()
        return render_template('admin.html', patients=patients, display="visible",␣
↪sel = 'p')
```

Admins can see all doctors on the admin page via this function:

```
#server.py

@app.route("/ShowAllUsers/",  methods=['GET', 'POST'])
def show_users():
        state = "SELECT ALL * FROM USERS"
        with dbapi2.connect(db_url) as connection:
                cursor = connection.cursor()
                cursor.execute(state)
                users = cursor.fetchall()
                cursor.close()
        return render_template('admin.html', users=users, display="visible", sel='u')
```

Admins who want to delete a patient can delete a patient via this function:

```
#server.py

@app.route("/del_patient/<int:patient_id>",  methods=['GET', 'POST'])
def del_patient(patient_id):
        state = "DELETE FROM PATIENT WHERE ID={}".format(patient_id)
        with dbapi2.connect(db_url) as connection:
                cursor = connection.cursor()
                cursor.execute(state)
                cursor.close()
        return redirect(url_for('show_patients'))
```

Admins who want to delete a user can delete him/her via this function:

```
#server.py
@app.route("/del_user/<int:user_id>",  methods=['GET', 'POST'])
def del_user(user_id):
        state = "DELETE FROM USERS WHERE ID={}".format(user_id)
        with dbapi2.connect(db_url) as connection:
                cursor = connection.cursor()
                cursor.execute(state)
                cursor.close()
        return redirect(url_for('show_users'))
```

Admins can exit the page via this function:

```
#server.py

@app.route("/Exit/",  methods=['GET', 'POST'])
def Exit():
        return render_template('home.html')
```

When doctors fills information about patients and clicks save button the patient added the system via this function:

```
#server.py

@app.route("/add_new_patient/", methods=['GET', 'POST'])
```

(continues on next page)

(continued from previous page)

```
def add_new_patient():
        bloody = request.form["blood"]
        blood_new = "null"

        if (bloody != ""):
                if bloody == "AB+":
                        blood_new = 1
                elif bloody == "AB-":
                        blood_new = 2
                elif bloody == "0+":
                        blood_new = 3
                elif bloody == "0-":
                        blood_new = 4
                elif bloody == "A+":
                        blood_new = 5
                elif bloody == "AB-":
                        blood_new = 6
                elif bloody == "B+":
                        blood_new = 7
                elif bloody == "B-":
                        blood_new = 8
                else:
                        blood_new = "null"
        statements = ["INSERT INTO PATIENT(NAME, AGE, WEIGHT, HEIGHT, LAST_
↪EXAMINATION_DATE, "
                                 "BLOOD_TYPE) VALUES('{}', {}, {}, {}, '{}', {})␣
↪RETURNING ID".format(request.form["name"], request.form["age"],
                                                                             ␣
↪                                                              request.form["weight"],
                                                                             ␣
↪                                                              request.form["height"],␣
↪request.form["exam_date"],
                                                                             ␣
↪                                                              blood_new)]
        with dbapi2.connect(db_url) as connection:
                cursor = connection.cursor()
                cursor.execute(statements[0])
                x = cursor.fetchone()
                cursor.close()
        statements = []
        if request.form["fam_dis"] != "":
                state = "INSERT INTO FAMILY_DISEASE(NAME, AREA, PERSON) VALUES('{}', '
↪{}', {})".format(request.form["fam_dis"],
                                                                             ␣
↪                                                                            ␣
↪                  request.form["fam_area"],
                                                                             ␣
↪                                                                            ␣
↪                  x[0])
                statements.append(state)
        if int(request.form["fam_dis_num"]) != 0:
                for i in range(int(request.form["fam_dis_num"])):
                        state = "INSERT INTO FAMILY_DISEASE(NAME, AREA, PERSON)␣
↪VALUES('{}', '{}', {})".format(request.form["fam_dis" + str(i)],
                                                                             ␣
↪                                                                            ␣
↪                          request.form["fam_area" + str(i)],
```

(continues on next page)

(continued from previous page)

```
                                                                                                   ␣
↪                                                                                                  ␣
↪                          x[0])
                        statements.append(state)
        if request.form["disco"] != "":
                state = "INSERT INTO DISCOMFORT(NAME, AREA, LEVELS, PERSON) VALUES('{}
↪', '{}', {}, {})".format(request.form["disco"],
                                                                                                   ␣
↪                                                                                                  ␣
↪                  request.form["disco_area"], request.form["disco_level"],
                                                                                                   ␣
↪                                                                                                  ␣
↪                  x[0])
                statements.append(state)
        if int(request.form["disco_number"]) != 0:
                for i in range(int(request.form["disco_number"])):
                        state = "INSERT INTO DISCOMFORT(NAME, AREA, LEVELS, PERSON)␣
↪VALUES('{}', '{}', {}, {})".format(request.form["disco" + str(i)],
                                                                                                   ␣
↪                                                                                                  ␣
↪                 request.form["disco_area" + str(i)], request.form["disco_level" +␣
↪str(i)], x[0])
                        statements.append(state)
        if request.form["med_dev"] != "":
                state = "INSERT INTO MEDICAL_DEVICE(NAME, AREA, PERSON) VALUES('{}', '
↪{}', {})".format(request.form["med_dev"],
                                                                                                   ␣
↪                                                                                                  ␣
↪                  request.form["med_dev_area"],
                                                                                                   ␣
↪                                                                                                  ␣
↪                  x[0])
                statements.append(state)
        if int(request.form["med_dev_number"]) != 0:
                for i in range(int(request.form["med_dev_number"])):
                        state = "INSERT INTO MEDICAL_DEVICE(NAME, AREA, PERSON)␣
↪VALUES('{}', '{}', {})".format(request.form["med_dev" + str(i)],
                                                                                                   ␣
↪                                                                                                  ␣
↪                          request.form["med_dev_area" + str(i)],
                                                                                                   ␣
↪                                                                                                  ␣
↪                          x[0])
                        statements.append(state)
        if request.form["medi"] != "":
                state = "INSERT INTO MEDICATION(NAME, USAGES, PERSON) VALUES('{}', '{}
↪', {})".format(request.form["medi"],
                                                                                                   ␣
↪                                                                                                  ␣
↪                  request.form["medi_area"],
                                                                                                   ␣
↪                                                                                                  ␣
↪                  x[0])
                statements.append(state)
        if int(request.form["medi_number"]) != 0:
                for i in range(int(request.form["medi_number"])):
                        state = "INSERT INTO MEDICATION(NAME, USAGES, PERSON) VALUES('
↪{}', '{}', {})".format(request.form["medi" + str(i)],
```

(continues on next page)

(continued from previous page)

```
                                                                                   ␣
↪                                                                                  ␣
↪                              request.form["medi_area" + str(i)],
                                                                                   ␣
↪                                                                                  ␣
↪                              x[0])
                            statements.append(state)
        if request.form["surge"] != "":
                state = "INSERT INTO SURGERY(NAME, AREA, LEVELS, PERSON) VALUES('{}',
↪'{}', {}, {})".format(request.form["surge"],
                                                                                   ␣
↪                                                                                  ␣
↪                      request.form["surge_area"], request.form["surge_level"],
                                                                                   ␣
↪                                                                                  ␣
↪                      x[0])
                statements.append(state)
        if int(request.form["surge_number"]) != 0:
                for i in range(int(request.form["surge_number"])):
                        state = "INSERT INTO SURGERY(NAME, AREA, LEVELS, PERSON)␣
↪VALUES('{}', '{}', {}, {})".format(request.form["surge" + str(i)],
                                                                                   ␣
↪                                                                                  ␣
↪                              request.form["surge_area" + str(i)], request.form["surge_
↪level" + str(i)],
                                                                                   ␣
↪                                                                                  ␣
↪                              x[0])
                            statements.append(state)
        if request.form["allergy"] != "":
                state = "INSERT INTO ALLERGY(NAME, AREA, PERSON) VALUES('{}', '{}', {}
↪)".format(request.form["allergy"],
                                                                                   ␣
↪                                                                                  ␣
↪                      request.form["allergy_area"],
                                                                                   ␣
↪                                                                                  ␣
↪                      x[0])
                statements.append(state)
        if int(request.form["allergy_number"]) != 0:
                for i in range(int(request.form["allergy_number"])):
                        state = "INSERT INTO ALLERGY(NAME, AREA, PERSON) VALUES('{}',
↪'{}', {})".format(request.form["allergy" + str(i)],
                                                                                   ␣
↪                                                                                  ␣
↪                              request.form["allergy_area" + str(i)],
                                                                                   ␣
↪                                                                                  ␣
↪                              x[0])
                            statements.append(state)

        with dbapi2.connect(db_url) as connection:
                cursor = connection.cursor()
                for state in statements:
                        cursor.execute(state)
                cursor.close()
```

(continues on next page)

(continued from previous page)

```
        return render_template('doctor.html', name="", age="", weight="", height="",
                                                         examinate_date="",␣
↪blood_type="", family_diseases="", discomforts="",
                                                         medications="",␣
↪surgeries="", medical_device="", allergies="", uw='n', display="none",
                                                         display_wei="none",␣
↪display_fam="none", uf='n', display_fam_ad="none",
                                                         display_fam_del="none
↪")
```

When doctors check a patient by id in the doctor page can see informations of the patient (if exist in the database) via this function:

```
#server.py

@app.route('/check/',  methods=['GET', 'POST'])
def check():
        global blood, age, name, weight, height, aller, med_dev, surge, medi, discomp,
↪ family_diseases, exam
        array = []
        global id
        id = request.form["id"]
        statements = ["SELECT WEIGHT FROM PATIENT WHERE ID={}".format(request.form["id
↪"]),
                                 "SELECT HEIGHT FROM PATIENT WHERE ID={}".
↪format(request.form["id"]),
                                 "SELECT NAME FROM PATIENT WHERE ID={}".
↪format(request.form["id"]),
                                 "SELECT AGE FROM PATIENT WHERE ID={}".format(request.
↪form["id"]),
                                 "SELECT NAME, AREA FROM ALLERGY WHERE PERSON={}".
↪format(request.form["id"]),
                                 "SELECT LAST_EXAMINATION_DATE FROM PATIENT WHERE ID=
↪{}".format(request.form["id"]),
                                 "SELECT BLOOD_TYPE FROM PATIENT WHERE ID={}".
↪format(request.form["id"]),
                                 "SELECT NAME, AREA, LEVELS FROM DISCOMFORT WHERE␣
↪PERSON={}".format(request.form["id"]),
                                 "SELECT NAME, USAGES FROM MEDICATION WHERE PERSON={}
↪".format(request.form["id"]),
                                 "SELECT NAME, AREA FROM MEDICAL_DEVICE WHERE PERSON=
↪{}".format(request.form["id"]),
                                 "SELECT NAME, AREA, LEVELS FROM SURGERY WHERE PERSON=
↪{}".format(request.form["id"]),
                                 "SELECT NAME, AREA FROM FAMILY_DISEASE WHERE PERSON=
↪{}".format(request.form["id"])
                                 ]

        with dbapi2.connect(db_url) as connection:
                cursor = connection.cursor()
                for state in statements:
                        cursor.execute(state)
                        array.append(cursor.fetchall())
                cursor.close()
                if(array[6] != [] ):
                        if (array[6][0][0] == 1):
```

(continues on next page)

(continued from previous page)

```
                                blood = "AB+"
                        elif (array[6][0][0] == 2):
                                blood = "AB-"
                        elif (array[6][0][0] == 3):
                                blood = "0+"
                        elif (array[6][0][0] == 4):
                                blood = "0-"
                        elif (array[6][0][0] == 5):
                                blood = "A+"
                        elif (array[6][0][0] == 6):
                                blood = "AB-"
                        elif (array[6][0][0] == 7):
                                blood = "B+"
                        elif (array[6][0][0] == 8):
                                blood = "B-"
                        else:
                                blood = ""
                                ############################################
                family_diseases = list()
                for i in array[11]:
                        family_diseases.append(i[0] + " area: " + i[1])
                discomp = list()
                for i in array[7]:
                        discomp.append(i[0] + " area: " + i[1] + " level: " +␣
↪str(i[2]))
                medi = list()
                for i in array[8]:
                        medi.append(i[0] + " usage: " + i[1])
                surge = list()
                for i in array[10]:
                        surge.append(i[0] + " area: " + i[1] + " level: " + str(i[2]))
                med_dev = list()
                for i in array[9]:
                        med_dev.append( i[0]  + " area: " + i[1])
                aller = list()
                for i in array[4]:
                        aller.append( i[0] + " area: " + i[1])

                        ##############################################################
↪#
                if (array[1] != []):
                        height = array[1][0][0]
                else:
                        height = "-"
                if (array[0] != []):
                        weight = array[0][0][0]
                else:
                        weight = "-"
                if (array[5] != []):
                        exam = array[5][0][0]
                else:
                        exam = "no_date"
        if (array[2] == []):
                return render_template('doctor.html', name="", age="", weight="",␣
↪height="",
                                                        examinate_date="", blood_type="",␣
↪family_diseases="", discomforts="", display_blood="none", upblood='n',
```

(continues on next page)

(continued from previous page)

```
                                                    medications="", surgeries="",␣
↪medical_device="", allergies="", uw='n', uphei='n', display_hei="none",
                                                    display="none", display_wei="none",
↪ display_fam="none", uf='n', display_date="none", up_exam_date="n", display_fam_ad=
↪"none",
                                                    display_fam_del="none", no_res=
↪"Patient didn't found!")
        else:
                name = array[2][0][0]
                age = array[3][0][0]
                return render_template('doctor.html', name=array[2][0][0],␣
↪age=array[3][0][0], weight=weight, height=height,
                                                    examinate_date=exam, blood_
↪type=blood, family_diseases=family_diseases, discomforts=discomp, updisco='n',
                                                            display_med_dev="none",␣
↪display_med_dev_ad="none", display_med_dev_del="none",
                                                            upmed_dev='n', display_
↪aller="none", display_aller_ad="none", display_aller_del="none", upaller='n',
                                                    medications=medi, surgeries=surge,␣
↪medical_device=med_dev, uphei='n', display_hei="none", display_blood="none",␣
↪upblood='n',
                                                    allergies=aller, display_date="none
↪", up_exam_date="n", uw='n', display="visible", display_wei="none", display_fam=
↪"none",
                                                    uf='n', display_fam_ad="none",␣
↪display_fam_del="none", display_disco="none", display_disco_ad="none", display_
↪disco_del="none", display_medi="none",
                                                    display_medi_ad="none", display_
↪medi_del="none", upmedi='n', display_surge="none", display_surge_ad="none", display_
↪surge_del="none", upsurge='n',)
```

Doctors can update weight, height, blood_type and last_examination_date properties of patients. Example function is placed below.

```
#server.py

@app.route("/Update_wei/", methods=['GET', 'POST'])
def update_wei():
        global count, height, weight, age, name, exam, blood, family_diseases,␣
↪discomp, medi, surge, med_dev, aller
        if (count == 0):
                count = 1
                return render_template('doctor.html', name=name, age=age,␣
↪weight=weight, height=height,
                                                                 examinate_date=exam,␣
↪blood_type=blood, family_diseases=family_diseases, discomforts=discomp,
                                                                 medications=medi,␣
↪surgeries=surge, medical_device=med_dev, allergies=aller, uw='y',
                                                            display_med_dev="none",␣
↪display_med_dev_ad="none", display_med_dev_del="none",
                                                            upmed_dev='n', display_
↪aller="none", display_aller_ad="none", display_aller_del="none", upaller='n',
                                                                 display="visible",␣
↪display_wei="visible", display_fam="none", uf='n', display_fam_ad="none",
                                                                 display_fam_del="none
↪", uphei='n', display_hei="none", display_blood="none", upblood='n',display_date=
↪"none", up_exam_date="n",
```

(continues on next page)

(continued from previous page)

```
                                                            display_disco="none",␣
↪display_disco_ad="none", display_disco_del="none", updisco='n', display_medi="none",
                                                        display_medi_ad="none",␣
↪display_medi_del="none", upmedi='n', display_surge="none", display_surge_ad="none",␣
↪display_surge_del="none", upsurge='n',)
        else :
                count = 0
                old_wei = weight
                weight = request.form["new_weight"]
                if (weight != ""):
                        state = "UPDATE PATIENT SET WEIGHT={} WHERE ID={}".
↪format(weight, id)
                        with dbapi2.connect(db_url) as connection:
                                cursor = connection.cursor()
                                cursor.execute(state)
                                cursor.close()
                else:
                        weight = old_wei
                return render_template('doctor.html', name=name, age=age,␣
↪weight=weight, height=height,
                                                            examinate_date=exam,␣
↪blood_type=blood, family_diseases=family_diseases, discomforts=discomp,
                                                            medications=medi,␣
↪surgeries=surge, medical_device=med_dev, allergies=aller, uw='n',
                                                        display_med_dev="none",␣
↪display_med_dev_ad="none", display_med_dev_del="none",
                                                        upmed_dev='n', display_
↪aller="none", display_aller_ad="none", display_aller_del="none", upaller='n',
                                                            display="visible",␣
↪display_wei="none", display_fam="none", uf='n', display_fam_ad="none",
                                                            display_fam_del="none
↪", uphei='n', display_hei="none", display_blood="none", upblood='n',display_date=
↪"none", up_exam_date="n",
                                                            display_disco="none",␣
↪display_disco_ad="none", display_disco_del="none", updisco='n', display_medi="none",
                                                        display_medi_ad="none",␣
↪display_medi_del="none", upmedi='n', display_surge="none", display_surge_ad="none",␣
↪display_surge_del="none", upsurge='n',)
```

Doctors can delete a familiy disease via the function below. Surgery, Medical Devices, Allergies, Medications and Discomforts elements are also deleted via similar functions.

```
#server.py

@app.route("/Delete_fam/", methods=['GET', 'POST'])
def delete_fam():
        global delete_fam_count, family_diseases
        if (delete_fam_count == 0):
                if family_diseases == "-" or family_diseases == []:
                        return render_template('doctor.html', name=name, age=age,␣
↪weight=weight, height=height,
                                                                examinate_
↪date=exam, blood_type=blood, family_diseases=family_diseases,
                                                                                ␣
↪discomforts=discomp, display_medi="none", display_medi_ad="none", display_medi_del=
↪"none", upmedi='n',
```

(continues on next page)

(continued from previous page)

```
                                                                       medications=medi,␣
↪surgeries=surge, medical_device=med_dev, updisco='n',
                                                                       allergies=aller,␣
↪uw='n', display="visible", display_wei="none", display_fam="none",
                                                                       uf='n', display_
↪fam_ad="none", display_blood="none", upblood='n',  uphei='n', display_hei="none",␣
↪display_date="none", up_exam_date="n",
                                                                       display_fam_del=
↪"none", display_disco="none", display_disco_ad="none", display_disco_del="none",
                                                                       display_med_dev=
↪"none", display_med_dev_ad="none", display_med_dev_del="none",
                                                                       upmed_dev='n',␣
↪display_aller="none", display_aller_ad="none", display_aller_del="none", upaller='n
↪',
                                                                       display_surge="none
↪", display_surge_ad="none", display_surge_del="none", upsurge='n')
                        else:
                                delete_fam_count = 1
                                return render_template('doctor.html', name=name, age=age,␣
↪weight=weight, height=height,
                                                                    examinate_date=exam,␣
↪blood_type=blood, family_diseases=family_diseases, discomforts=discomp,
                                                                       display_med_dev=
↪"none", display_med_dev_ad="none", display_med_dev_del="none",
                                                                       upmed_dev='n',␣
↪display_aller="none", display_aller_ad="none", display_aller_del="none", upaller='n
↪',
                                                                    medications=medi,␣
↪surgeries=surge, medical_device=med_dev, allergies=aller, uw='n',
                                                                    display="visible",␣
↪display_wei="none", display_fam="none", uf='n', display_fam_ad="none",
                                                                       display_blood="none
↪", upblood='n', uphei='n', display_hei="none",
                                                                       display_date="none
↪", up_exam_date="n", updisco='n', display_medi="none", display_medi_ad="none",␣
↪display_medi_del="none", upmedi='n',
                                                                    display_fam_del=
↪"visible", display_disco="none", display_disco_ad="none", display_disco_del="none",
                                                                       display_surge="none
↪", display_surge_ad="none", display_surge_del="none", upsurge='n')
                else :
                        delete_fam_count = 0
                        statement = []
                        for x in request.form.getlist("OK"):
                                y = x.split(" area: ")
                                statement.append("DELETE FROM FAMILY_DISEASE WHERE NAME='{}'".
↪format(y[0]))
                                family_diseases.remove(x)
                        with dbapi2.connect(db_url) as connection:
                                cursor = connection.cursor()
                                for state in statement:
                                        cursor.execute(state)
                                cursor.close()
                        return render_template('doctor.html', name=name, age=age,␣
↪weight=weight, height=height,
                                                                examinate_date=exam, blood_
↪type=blood, family_diseases=family_diseases,
```

(continues on next page)

(continued from previous page)

```
                                                           discomforts=discomp,␣
↪display_blood="none", upblood='n',  uphei='n', display_hei="none", display_date=
↪"none", up_exam_date="n",
                                                           medications=medi,␣
↪surgeries=surge, medical_device=med_dev, allergies=aller, uw='n',
                                                           display="visible", display_
↪wei="none", display_fam="none", uf='n', display_fam_ad="none",
                                                           display_surge="none",␣
↪display_surge_ad="none", display_surge_del="none", upsurge='n',
                                                           display_med_dev="none",␣
↪display_med_dev_ad="none", display_med_dev_del="none",
                                                           upmed_dev='n', display_
↪aller="none", display_aller_ad="none", display_aller_del="none", upaller='n',
                                                           display_disco="none",␣
↪display_disco_ad="none", display_disco_del="none",
                                                           display_fam_del="none",␣
↪updisco='n', display_medi="none", display_medi_ad="none", display_medi_del="none",␣
↪upmedi='n')
```

Doctors can update a familiy disease via the method below. Surgery, Medical Devices, Allergies, Medications and Discomforts elements are also updated via similar functions.

```
#server.py

@app.route("/Update_fam/", methods=['GET', 'POST'])
def update_fam():
        global count_fam, family_diseases
        if (count_fam == 0):
                count_fam = 1
                return render_template('doctor.html', name=name, age=age,␣
↪weight=weight, height=height,
                                                                examinate_date=exam,␣
↪blood_type=blood, family_diseases=family_diseases, discomforts=discomp,
                                                                medications=medi,␣
↪surgeries=surge, medical_device=med_dev, allergies=aller, uw='n',
                                                           display_med_dev="none",␣
↪display_med_dev_ad="none", display_med_dev_del="none",
                                                           upmed_dev='n', display_
↪aller="none", display_aller_ad="none", display_aller_del="none", upaller='n',
                                                                display="visible",␣
↪display_wei="none", display_fam="visible", uf='y', display_fam_ad="none",
                                                                display_blood="none",␣
↪upblood='n', uphei='n', display_hei="none", display_date="none", updisco='n',
                                                                up_exam_date="n",␣
↪display_fam_del="none", display_disco="none", display_disco_ad="none", display_
↪disco_del="none",
                                                                display_surge = "none
↪", display_surge_ad = "none", display_surge_del = "none", upsurge = 'n',
                                                                display_medi="none",␣
↪display_medi_ad="none", display_medi_del="none", upmedi='n')
        else :
                count_fam = 0
                i = 0
                statement = []
                if family_diseases == "-" or family_diseases == []:
                        if (request.form["fam_dis"] != ""):
```

(continues on next page)

(continued from previous page)

```
                                family_diseases.append(request.form["fam_dis"] + "␣
↪area: " + request.form["area"])
                                ad = id
                                statement.append("INSERT INTO FAMILY_DISEASE(NAME,␣
↪AREA, PERSON) VALUES('{}', '{}', {})".format(request.form["fam_dis"],
                                                                                    ␣
↪                                                                                   ␣
↪                               request.form["area"], ad))
                else:
                        for fam in family_diseases:
                                x = fam.split(' area: ')
                                if (request.form[x[0]] != ""):
                                        old_name = x[0]
                                        family_diseases[i] = (request.form[x[0]] + "␣
↪area: " + request.form[x[1]])
                                        statement.append("UPDATE FAMILY_DISEASE SET␣
↪NAME='{}', AREA='{}' WHERE NAME='{}'".format(request.form[x[0]], request.form[x[1]],
↪ old_name))
                                i+=1

                with dbapi2.connect(db_url) as connection:
                        cursor = connection.cursor()
                        for state in statement:
                                cursor.execute(state)
                        cursor.close()
                return render_template('doctor.html', name=name, age=age,␣
↪weight=weight, height=height,
                                                        examinate_date=exam,␣
↪blood_type=blood, family_diseases=family_diseases, discomforts=discomp,
                                                        medications=medi,␣
↪surgeries=surge, medical_device=med_dev, allergies=aller, uw='n',
                                                        display="visible",␣
↪display_wei="none", display_fam="none", uf='n', display_fam_ad="none",
                                                display_blood="none",␣
↪upblood='n', uphei='n', display_hei="none", display_date="none",
                                                display_med_dev="none",␣
↪display_med_dev_ad="none", display_med_dev_del="none",
                                                upmed_dev='n', display_
↪aller="none", display_aller_ad="none", display_aller_del="none", upaller='n',
                                                up_exam_date="n", display_
↪disco="none", display_disco_ad="none", display_disco_del="none",
                                                display_surge="none",␣
↪display_surge_ad="none", display_surge_del="none", upsurge='n',
                                                        display_fam_del="none
↪", updisco='n', display_medi="none", display_medi_ad="none", display_medi_del="none
↪", upmedi='n')
```

Doctors can add a familiy disease via the function below. Surgery, Medical Devices, Allergies, Medications and Discomforts elements are also added via similar functions.

```
#server.py

@app.route("/Add_fam/", methods=['GET', 'POST'])
def add_fam():
        global count_add_fam, family_diseases
        if count_add_fam == 0:
```

(continues on next page)

(continued from previous page)

```
                count_add_fam = 1
                return render_template('doctor.html', name=name, age=age,␣
↪weight=weight, height=height,
                                                examinate_date=exam, blood_
↪type=blood, family_diseases=family_diseases,
                                                discomforts=discomp,␣
↪medications=medi, surgeries=surge, medical_device=med_dev,
                                                allergies=aller, uw='n',␣
↪display="visible", display_wei="none", display_fam="none",
                                                display_blood="none",␣
↪upblood='n', uphei='n', display_hei="none", display_date="none",
                                                up_exam_date="n", display_
↪disco="none", display_disco_ad="none", display_disco_del="none",
                                                uf='n', display_fam_ad=
↪"visible", display_fam_del="none", updisco='n',
                                                display_med_dev="none",␣
↪display_med_dev_ad="none", display_med_dev_del="none",
                                                upmed_dev='n', display_
↪aller="none", display_aller_ad="none", display_aller_del="none", upaller='n',
                                                display_surge="none",␣
↪display_surge_ad="none", display_surge_del="none", upsurge='n',
                                                display_medi="none",␣
↪display_medi_ad="none", display_medi_del="none", upmedi='n')
        else:
                count_add_fam = 0
                if request.form["new_fam"] != "":
                        family_diseases.append(request.form["new_fam"] + " area: " +␣
↪request.form["fam_area"])
                        ad = id
                        state = "INSERT INTO FAMILY_DISEASE(NAME, AREA, PERSON)␣
↪VALUES('{}', '{}', {})".format(request.form["new_fam"], request.form["fam_area"],␣
↪ad)
                        with dbapi2.connect(db_url) as connection:
                                cursor = connection.cursor()
                                cursor.execute(state)
                                cursor.close()
                return render_template('doctor.html', name=name, age=age,␣
↪weight=weight, height=height,
                                                examinate_date=exam, blood_
↪type=blood, family_diseases=family_diseases,
                                                discomforts=discomp,␣
↪medications=medi, surgeries=surge, medical_device=med_dev,
                                                allergies=aller, uw='n',␣
↪display="visible", display_wei="none", display_fam="none",
                                                display_blood="none",␣
↪upblood='n', uphei='n', display_hei="none", display_date="none",
                                                up_exam_date="n", uf='n',␣
↪display_fam_ad="none", display_fam_del="none",
                                                display_med_dev="none",␣
↪display_med_dev_ad="none", display_med_dev_del="none",
                                                upmed_dev='n', display_
↪aller="none", display_aller_ad="none", display_aller_del="none", upaller='n',
                                                display_disco="none",␣
↪display_disco_ad="none", display_disco_del="none", updisco='n',
                                                display_surge="none",␣
↪display_surge_ad="none", display_surge_del="none", upsurge='n',
                                                display_medi="none",␣
↪display_medi_ad="none", display_medi_del="none", upmedi='n')
```

(continues on next page)

(continued from previous page)

2.4 Jinja:

To show buttons and contents of weight property we used the method below. Also contents of Examination date, Blood_type, Height properties implemented similar methods.

To show buttons and contents of familiy_diseases property we used the method below. Also contents of Allergy, Discomfort, Medical_device, Medication and Surgery properties implemented similar methods.

2.5 Ajax:

To add new line boxes in some properties via plus and minus buttons, that script is used.

```
#add_patient.html

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></
↪script>
<script>
        var disease_number = 0;
        $('#family_disease_btn').click(function (){
                let disease_html = `<li id=\"fam_dis${disease_number}\"><label><b>
↪Family Disease:&nbsp;</b></label><input type=\"text\" name=\"fam_dis${disease_
↪number}\"><label><b>&nbsp;Area:&nbsp;</b></label><input type=\"text\" name=\"fam_
↪area${disease_number}\"></li>`
                $('#family_disease_tr li:last').after(disease_html);
                disease_number ++;
                document.getElementById(`fam_dis_num`).value = disease_number;
        });

        $('#family_disease_remove').click(function (){
                disease_number --;
                document.getElementById(`fam_dis_num`).value = disease_number;
                var elem = document.getElementById(`fam_dis${disease_number}`);
                elem.parentNode.removeChild(elem);
                return false;
        });
        var disco_number = 0;
        $('#discomforts_btn').click(function (){
                let disco_html = `<li id=\"disco${disco_number}\"><label><b>
↪Discomfort:&nbsp;</b></label><input type=\"text\" name=\"disco${disco_number}\">
↪<label><b>&nbsp;Area:&nbsp;</b></label><input type=\"text\" name=\"disco_area$
↪{disco_number}\"><label><b>&nbsp;Level:&nbsp;</b></label><input type=\"number\"␣
↪name=\"disco_level${disco_number}\"></li>`
                $('#discomforts_tr li:last').after(disco_html);
                disco_number ++;
                document.getElementById(`disco_number`).value = disco_number;
        });

        $('#discomforts_remove').click(function (){
                disco_number --;
                document.getElementById(`disco_number`).value = disco_number;
```

(continues on next page)

(continued from previous page)

```
                var elem = document.getElementById(`disco${disco_number}`);
                elem.parentNode.removeChild(elem);
                return false;
        });
        var medi_number = 0;
        $('#medication_btn').click(function (){
                let medi_html = `<li id=\"medi${medi_number}\"><label><b>Medication:&
↪nbsp;</b></label><input type=\"text\" name=\"medi${medi_number}\"><label><b>&nbsp;
↪Usage:&nbsp;</b></label><input type=\"text\" name=\"medi_area${medi_number}\"></li>`
                $('#medication_tr li:last').after(medi_html);
                medi_number ++;
                document.getElementById(`medi_number`).value = medi_number;
        });

        $('#medication_remove').click(function (){
                medi_number --;
                document.getElementById(`medi_number`).value = medi_number;
                var elem = document.getElementById(`medi${medi_number}`);
                elem.parentNode.removeChild(elem);
                return false;
        });

        var med_dev_number = 0;
        $('#medical_device_btn').click(function (){
                let med_dev_html = `<li id=\"med_dev${med_dev_number}\"><label><b>
↪Medical Device:&nbsp;</b></label><input type=\"text\" name=\"med_dev${med_dev_
↪number}\"><label><b>&nbsp;Area:&nbsp;</b></label><input type=\"text\" name=\"med_
↪dev_area${med_dev_number}\"></li>`
                $('#medical_devices_tr li:last').after(med_dev_html);
                med_dev_number ++;
                document.getElementById(`med_dev_number`).value = med_dev_number;
        });

        $('#medical_device_remove').click(function (){
                med_dev_number --;
                document.getElementById(`med_dev_number`).value = med_dev_number;
                var elem = document.getElementById(`med_dev${med_dev_number}`);
                elem.parentNode.removeChild(elem);
                return false;
        });

        var surge_number = 0;
        $('#surge_btn').click(function (){
                let surge_html = `<li id=\"surge${surge_number}\"><label><b>Surgery:&
↪nbsp;</b></label><input type=\"text\" name=\"surge${surge_number}\"><label><b>&nbsp;
↪Area:&nbsp;</b></label><input type=\"text\" name=\"surge_area${surge_number}\">
↪<label><b>&nbsp;Level:&nbsp;</b></label><input type=\"number\" name=\"surge_level$
↪{surge_number}\"></li>`
                $('#surgeries_tr li:last').after(surge_html);
                surge_number ++;
                document.getElementById(`surge_number`).value = surge_number;
        });

        $('#surge_remove').click(function (){
                surge_number --;
                document.getElementById(`surge_number`).value = surge_number;
                var elem = document.getElementById(`surge${surge_number}`);
```

(continues on next page)

(continued from previous page)

```
                elem.parentNode.removeChild(elem);
                return false;
        });

        var allergy_number = 0;
        $('#allergy_btn').click(function (){
                let allergy_html = `<li id=\"allergy${allergy_number}\"><label><b>
↪Allergy:&nbsp;</b></label><input type=\"text\" name=\"allergy${allergy_number}\">
↪<label><b>&nbsp;Area:&nbsp;</b></label><input type=\"text\" name=\"allergy_area$
↪{allergy_number}\"></li>`
                $('#allergies_tr li:last').after(allergy_html);
                allergy_number ++;
                document.getElementById(`allergy_number`).value = allergy_number;
        });

        $('#allergy_remove').click(function (){
                allergy_number --;
                document.getElementById(`allergy_number`).value = allergy_number;
                var elem = document.getElementById(`allergy${allergy_number}`);
                elem.parentNode.removeChild(elem);
                return false;
        });
</script>
```

2.5.1 User, Discomfort, Medication, Family_diseases, Blood_type tables and related functions, html Implemented by Doğu Ozan Kumru

2.5.2 Patients, Allergy, Surgery, Medical_device tables and related functions Implemented by Talha Çomak