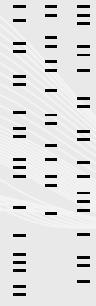
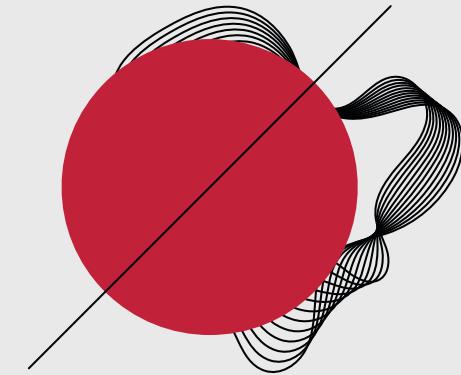
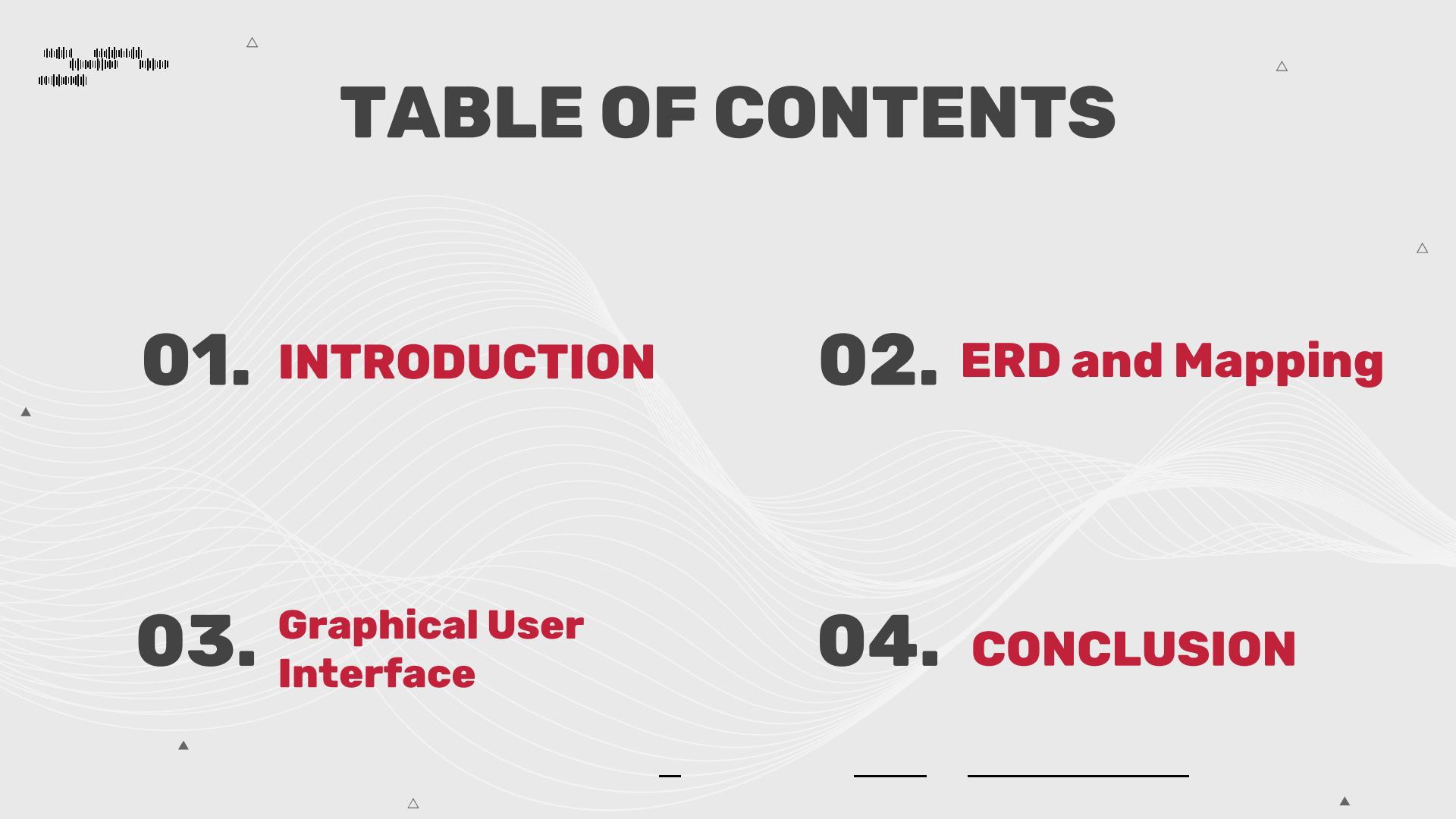


# COVID-19

# Management System



# TABLE OF CONTENTS



**01. INTRODUCTION**

**02. ERD and Mapping**

**03. Graphical User  
Interface**

**04. CONCLUSION**

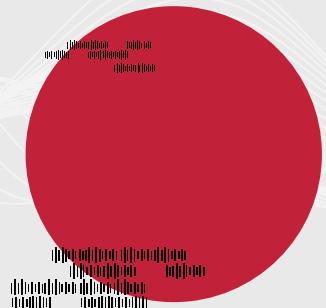
# 01.

## INTRODUCTION

The spread of corona is increasing dramatically, so diseases such as diabetes and cardiovascular disease have become a major problem for those who suffer from it, medical care and health systems. Therefore, we have created a website to help the patient, which is a set of medical services provided to patients and medical staff to facilitate the implementation of operations in the health and administrative services of the organization, speed up the system, and speed up the process of dealing with the crisis.

# Project goal

The goal of this project is to create a unified platform that provides the user with various operations that can be carried out, starting from reviewing the latest developments in the Corona virus pandemic, up to booking analyzes and serums, conducting medical consultations and following up on various operations.

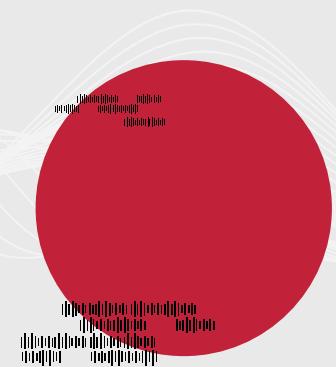


# Project Overview

Project aims to provide an appropriate and easy-to-use environment through which some services related to the management of the emerging corona virus crisis are provided, and these services are provided to users in various different places and environments.

The system provides its services to users through a platform supported by a database and a technical team.

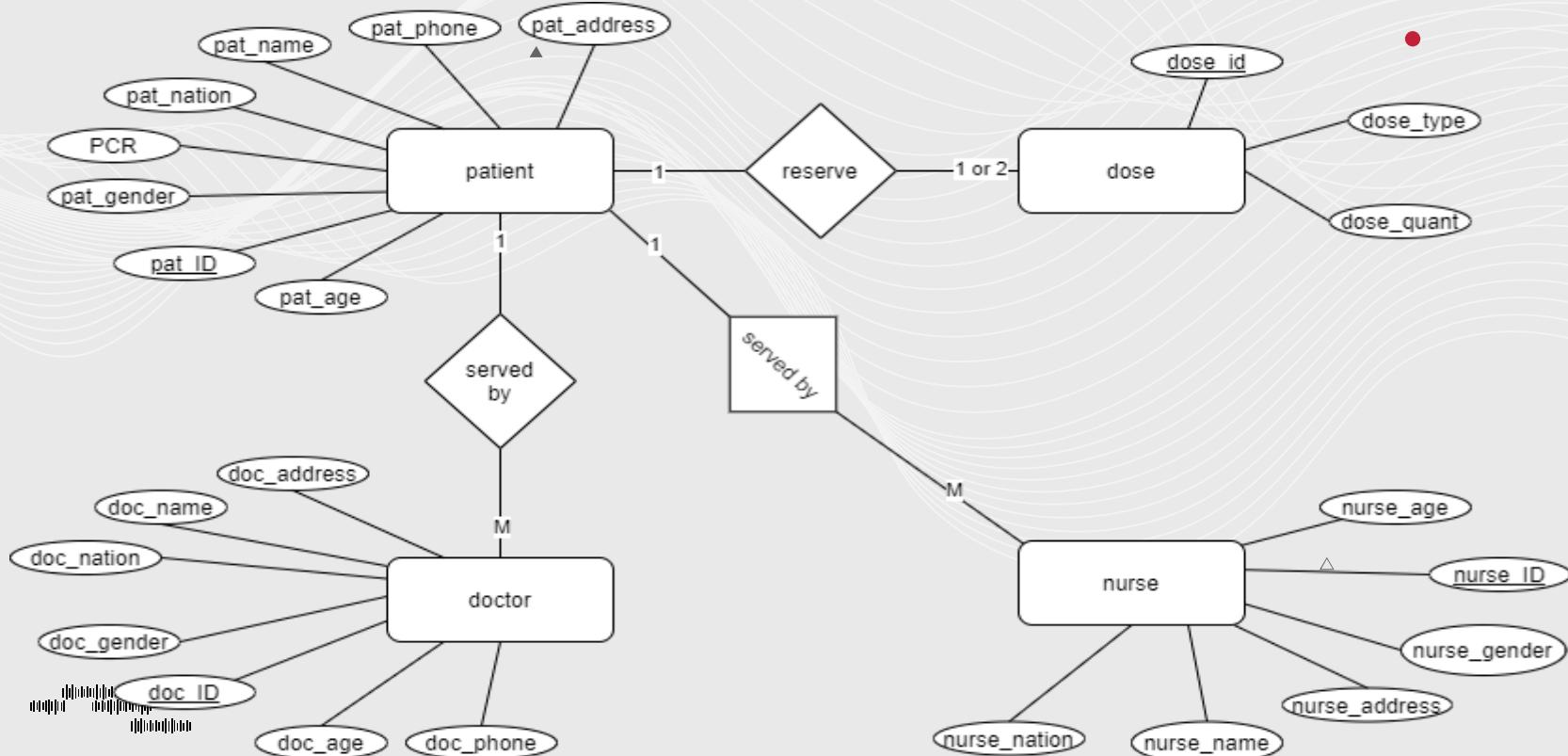
The user deals with the platform through a graphical interface, in order to facilitate dealing with the platform and perform various operations in a simple and easy way, and capable of serving all types of users (supervisor, doctor, patient) .



# 02.

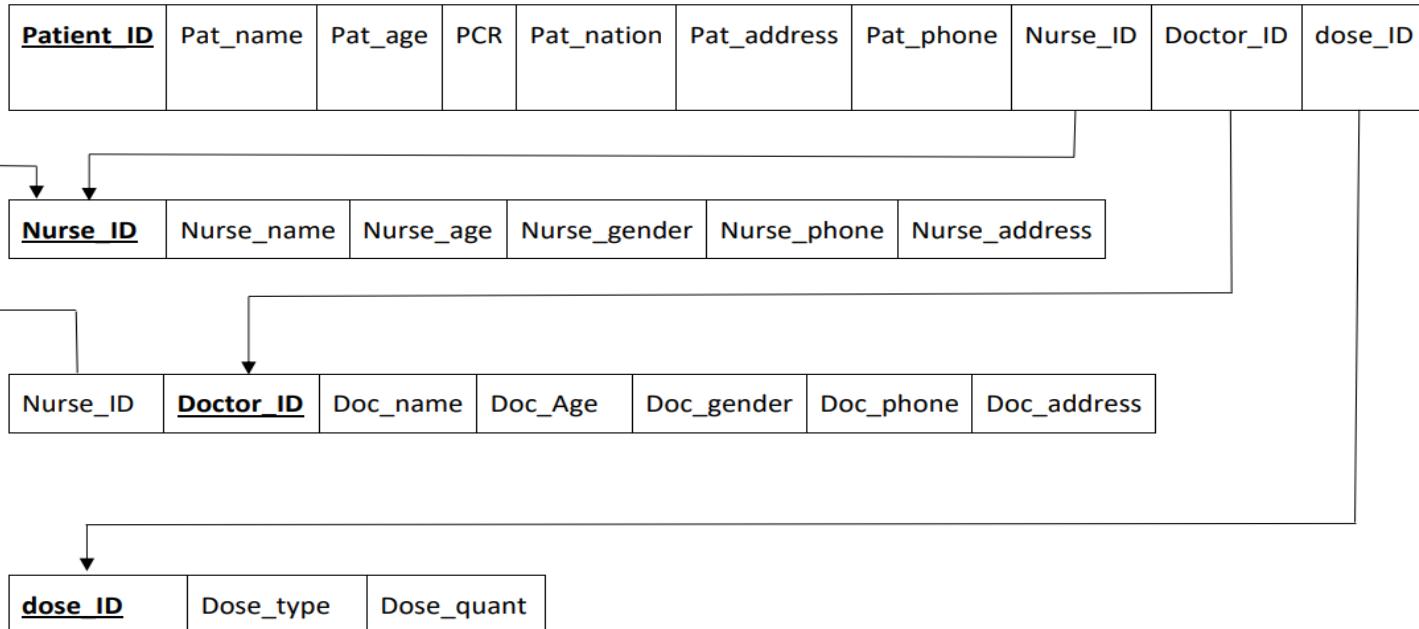
## ERD and Mapping

# Entity Relationship Diagram



# Mapping

## MAP



03.

# Graphical User Interface

## **Our Desktop Application consists of :**

- Login.
- Register.
- Home page For Patient.
  - Edit profile.
  - Reserve Does, Reserve PCR.
- Home page For Admin.
  - Add New Doctor
  - Add New Nurse
  - Add New Result
  - Show, update and delete all doctor
  - Show, update and delete all Nurse
  - Show, update and delete all patient

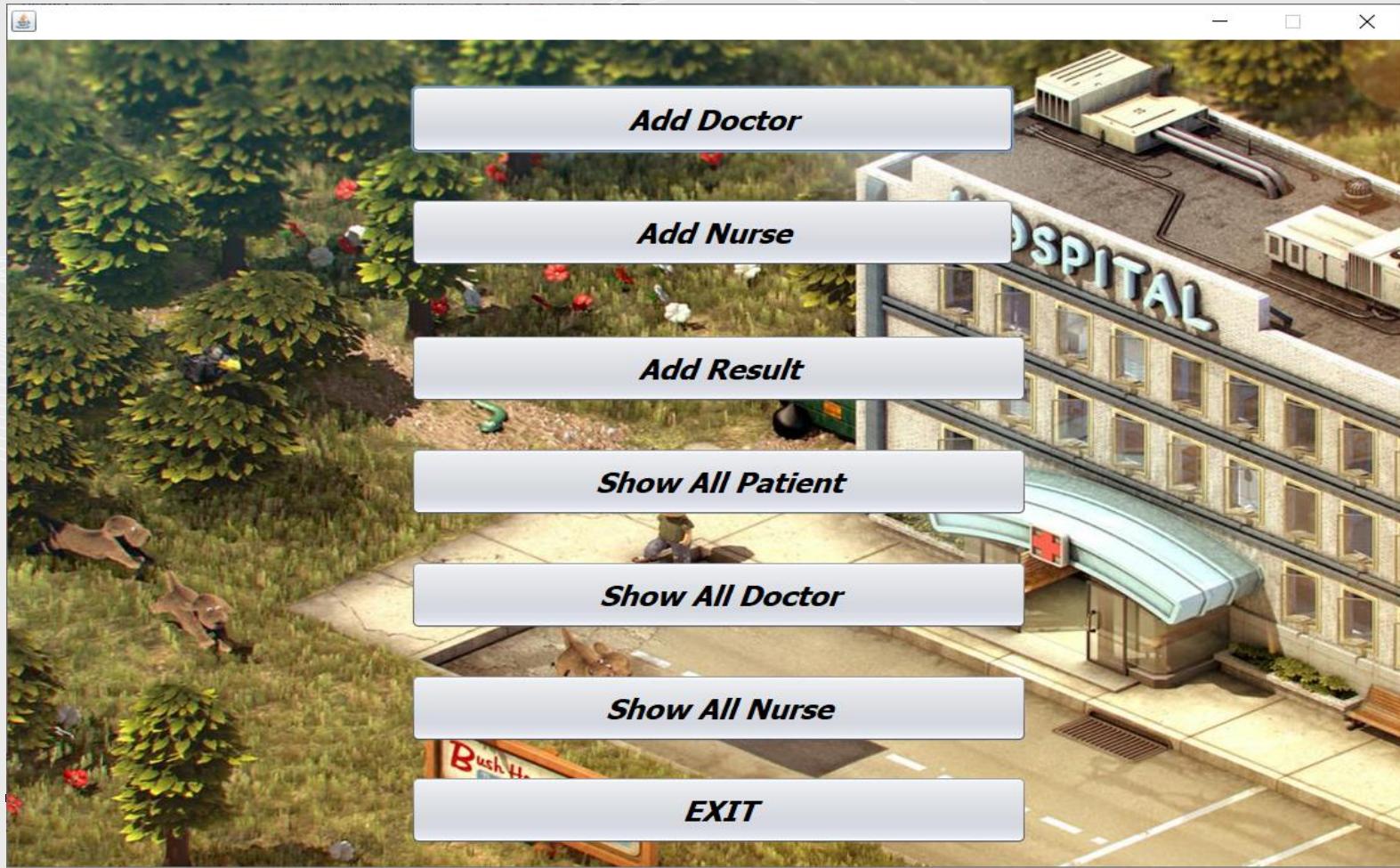


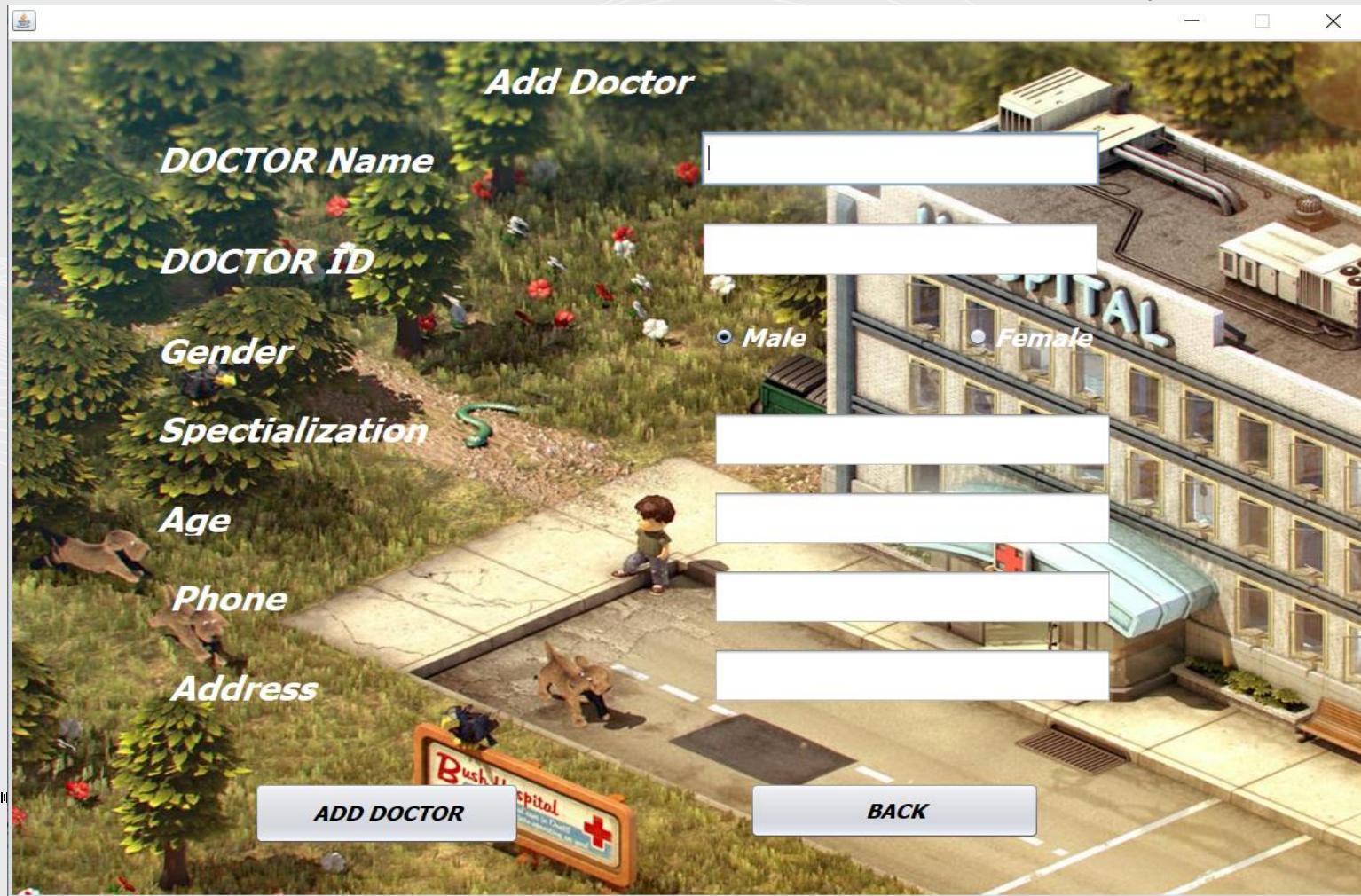














— □ ×

## Add Doctor

**DOCTOR Name**

Khaled

**DOCTOR ID**

\*\*\*\*\*

**Gender**

Male

Female

**Specialization**

Neurologist

**Age**

26

**Phone**

01279827842

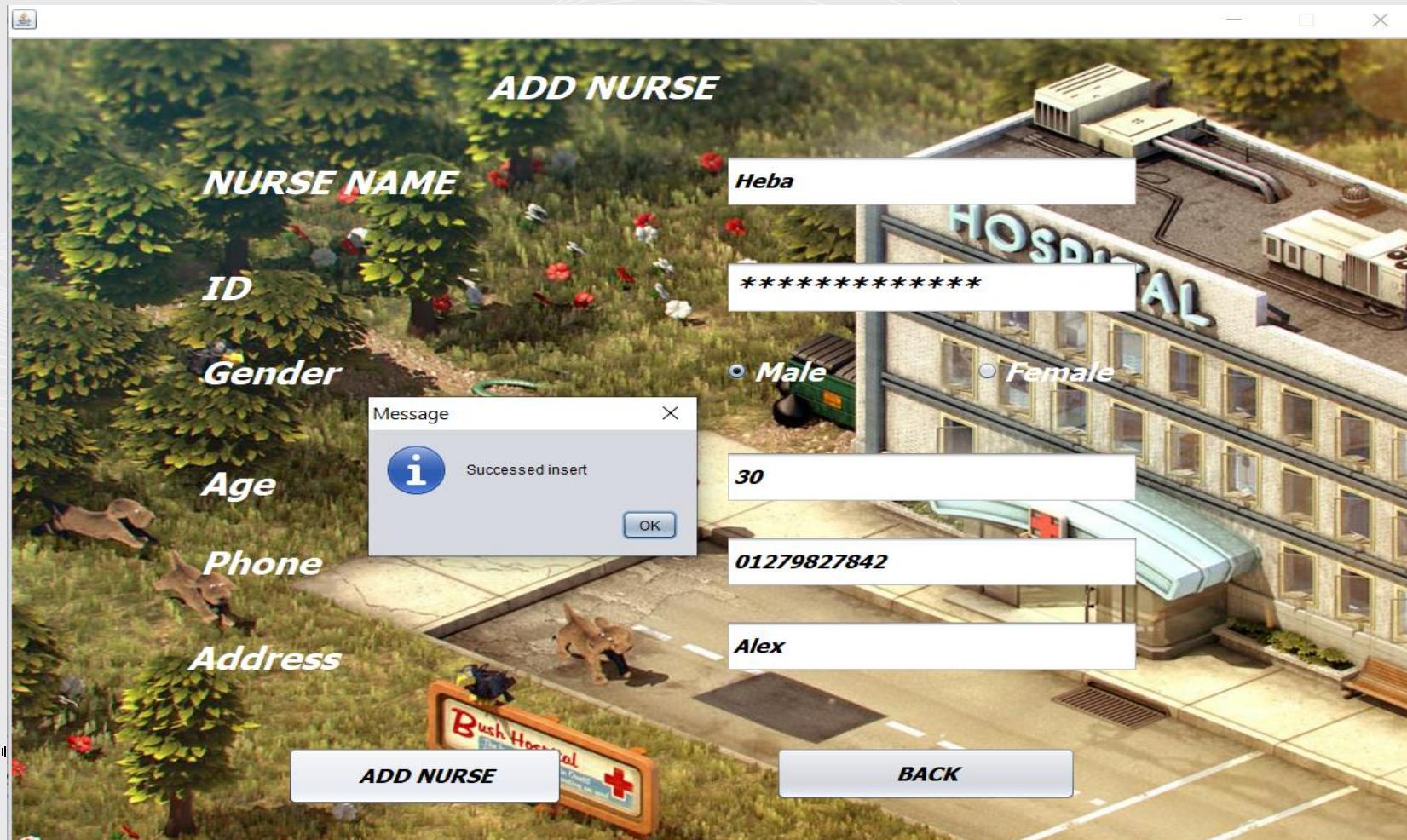
**Address**

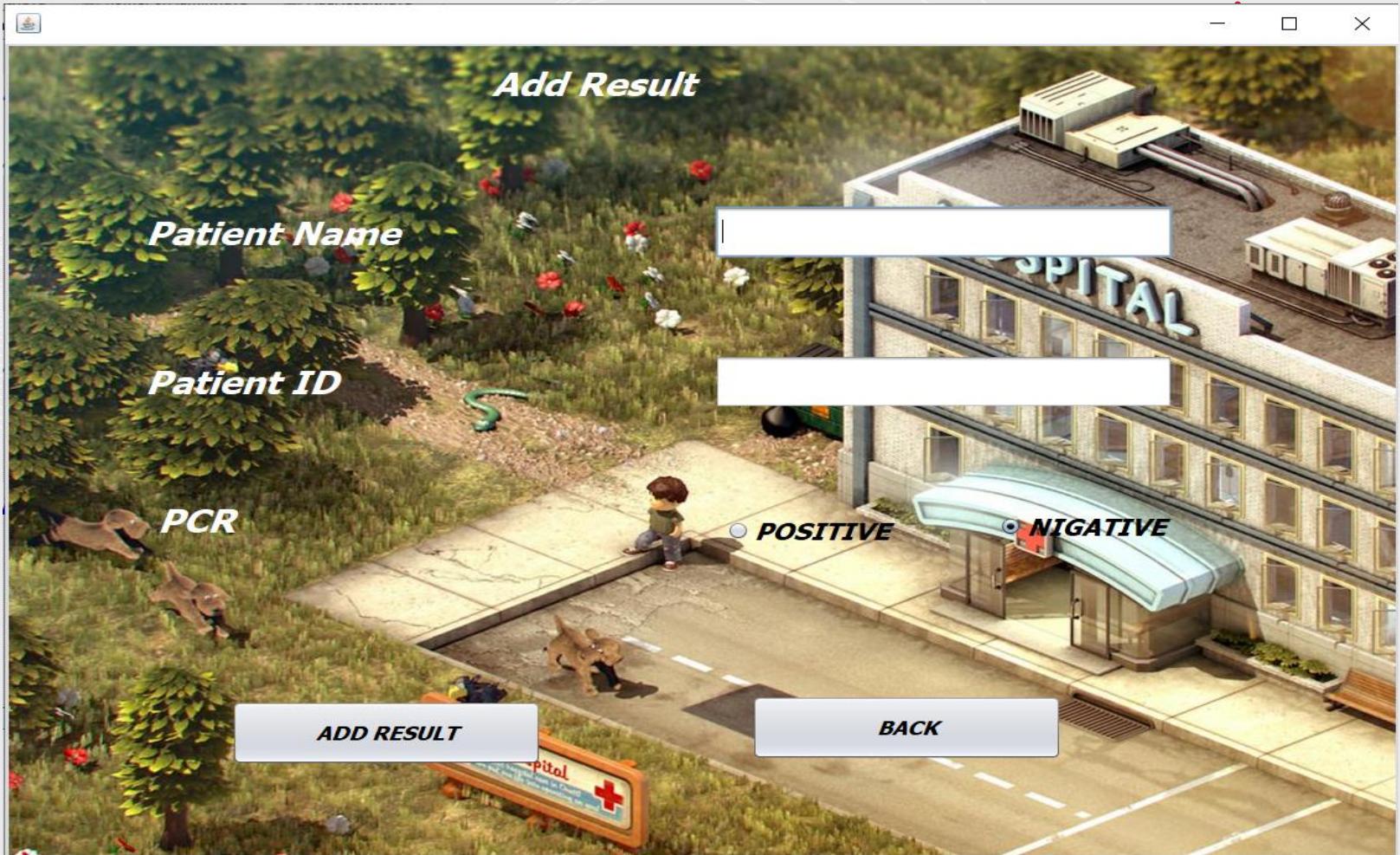
Benha

**ADD DOCTOR**

**BACK**









## Add Result

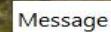
Patient Name

Ahmed

Patient ID

\*\*\*\*\*

PCR



Add Result Successfull

OK

POSITIVE

NIGATIVE

ADD RESULT

BACK

**DOCTOR NAME**

**DOCTOR ID**

**DOCTOR GENDER**

**DOCTOR Age**

**Specialization**

**DOCTOR phone**

**ADDRESS**

Male  Female

ID	Name	Nation ID	Specialization	Gender	Age	Phone	Address
1	Ahmed	30107151401...	Pediatrician	Male	30	1279827842	Benha
2	Taha	30105191826...	Endocrinologist	Male	32	1004651157	Alex
3	Rawan	30107986345...	Neurologist	Female	26	1001918125	Benha
4	khaled	30107151401...	Pediatrician	Male	30	1279827842	tanta
5	omar	30105191826...	Endocrinologist	Male	32	1004651157	monia
6	Heba	30107986345...	Neurologist	Female	26	1001918125	Assuit

**Update** **Delete** **Back**

**DOCTOR NAME**

**DOCTOR ID**

**DOCTOR GENDER**

**DOCTOR Age**

**Specialization**

**DOCTOR phone**

**ADDRESS**

**Rawan**

**30107986345287**

Male  Female

**26**

**Neurologist**

**1001918125**

**Benha**

ID	Name	Nation ID	Specialization	Gender	Age	Phone	Address
1	Ahmed	30107151401515	Pediatrician	Male	30	1279827842	Benha
2	Taha	30105191826374	Endocrinologist	Male	32	1004651157	Alex
3	Rawan	30107986345287	Neurologist	Female	26	1001918125	Benha
4	khaled	30107151401515	Pediatrician	Male	30	1279827842	tanta
5	omar	30105191826374	Endocrinologist	Male	32	1004651157	monfia
6	Heba	30107986345287	Neurologist	Female	26	1001918125	Assuit

**Update** **Delete** **Back**

**DOCTOR NAME**

**DOCTOR ID**

**DOCTOR GENDER**

**DOCTOR Age**

**Speclization**

**DOCTOR phone**

**ADDRESS**

**Message**

Successed updated

**khaled**

**30107151401515**

Male       Female

**32**

**Pediatrician**

**1279827842**

**tanta**

ID	Name	Nation ID	Specialization	Gender	Age	Phone	Address
1	Ahmed	30107151401515	Pediatrician	Male	30	1279827842	Benha
2	Taha	30105191826374	Endocrinologist	Male	32	1004651157	Alex
3	Rawan	30107986345287	Neurologist	Female	26	1001918125	Benha
4	<b>khaled</b>	<b>30107151401515</b>	<b>Pediatrician</b>	<b>Male</b>	<b>30</b>	<b>1279827842</b>	<b>tanta</b>
5	omar	30105191826374	Endocrinologist	Male	32	1004651157	monfia
6	Heba	30107986345287	Neurologist	Female	26	1001918125	Assuit

**Update**

**Delete**

**Back**

**DOCTOR NAME**

**DOCTOR ID**

**DOCTOR GENDER**

**DOCTOR Age**

**Specatilization**

**DOCTOR phone**

**ADDRESS**

**Message**

Successed deleted

OK

ID	Name	Nation ID	Specialization	Gender	Age	Phone	Address
1	Ahmed	30107151401515	Pediatrician	Male	30	1279827842	Benha
2	Taha	30105191826374	Endocrinologist	Male	32	1004651157	Alex
3	Rawan	30107986345287	Neurologist	Female	26	1001918125	Benha
4	Khaled	30107151401515	Pediatrician	Male	32	1279827842	tanta
5	omar	30105191826374	Endocrinologist	Male	32	1004651157	monfia
6	Heba	30107986345287	Neurologist	Female	26	1001918125	Assuit

**Update**

**Delete**

**Back**

omar

30105191826374

Male       Female

32

Endocrinologist

1004651157

monfia

**NURSE NAME**

**NATION ID**

**NURSE GENDER**

**NURSE Age**

**NURSE phone**

**Address**

ID	Name	Nation ID	Gender	Age	Phone	Address
1	omar	30107151401515	Male	30	01279827842	cairo
3	Rawan	30107986345287	Femal	26	01001918125	Benha
4	Fatma	30107986345287	Femal	26	01001918125	monfia
5	Rana	30107151401515	Femal	26	01279827842	shobra
6	heba	30107150151415	Femal	29	01279527862	cairo
7	reham	30105687145621	Femal	30	01005689432	Aswan
8	Heba	3010715140151	Male	30	01279827842	Alex

**Update**      **Delete**      **Back**

The image shows a mobile application window with a semi-transparent overlay. The overlay contains fields for entering a nurse's name, nation ID, gender, age, phone number, and address. It also displays a table of nurses with columns for ID, Name, Nation ID, Gender, Age, Phone, and Address. The row for 'Fatma' is highlighted. At the bottom are buttons for Update, Delete, and Back.

**NURSE NAME**

**NATION ID**

**NURSE GENDER**

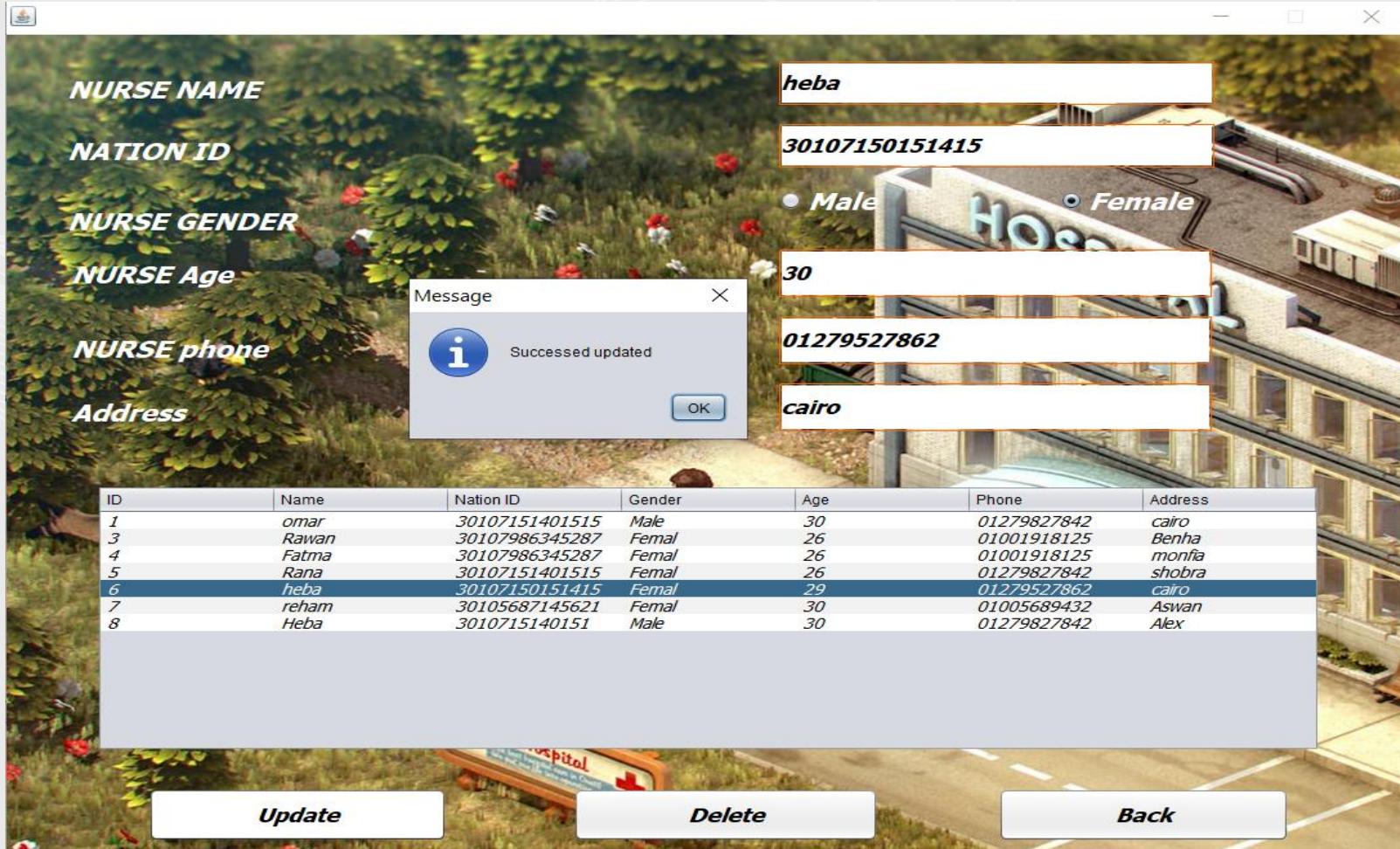
**NURSE Age**

**NURSE phone**

**Address**

ID	Name	Nation ID	Gender	Age	Phone	Address
1	omar	30107151401515	Male	30	01279827842	cairo
3	Rawan	30107986345287	Femal	26	01001918125	Benha
4	<b>Fatma</b>	<b>30107986345287</b>	<b>Femal</b>	<b>26</b>	<b>01001918125</b>	<b>monfia</b>
5	Rana	30107151401515	Femal	26	01279827842	shobra
6	heba	30107150151415	Femal	29	01279527862	cairo
7	reham	30105687145621	Femal	30	01005689432	Aswan
8	Heba	3010715140151	Male	30	01279827842	Alex

**Update**    **Delete**    **Back**





**NURSE NAME**

**NATION ID**

**NURSE GENDER**

**NURSE Age**

**NURSE phone**

**Address**

**reham**

**30105687145621**

Male     Female

**30**

**01005689432**

**Aswan**

Message ×

i Successed deleted

OK

ID	Name	Nation ID	Gender	Age	Phone	Address
1	omar	30107151401515	Male	30	01279827842	cairo
3	Rawan	30107986345287	Femal	26	01001918125	Benha
4	Fatma	30107986345287	Femal	26	01001918125	monfa
5	Rana	30107151401515	Femal	26	01279827842	shobra
6	heba	30107150151415	Femal	30	01279527862	cairo
7	reham	30105687145621	Femal	30	01005689432	Aswan
8	Heba	3010715140151	Male	30	01279827842	Alex

**Update**    **Delete**    **Back**

**PATIENT NAME**

**NATION ID**

**PATIENT GENDER**

**Patient Age**

**Patient phone**

**Patient PCR**

**Patient Does**

**Address**

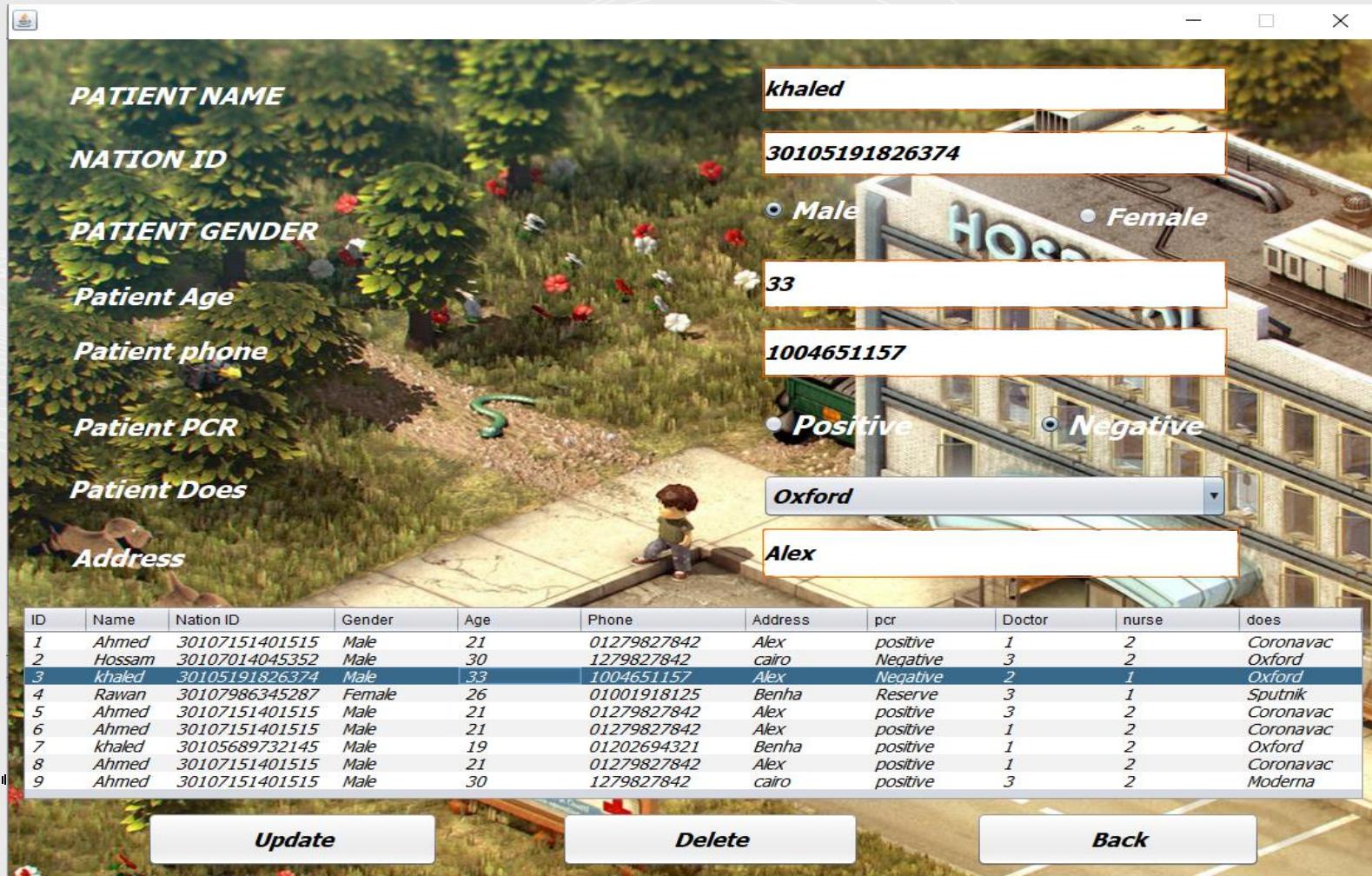
**Patient Name**:

**Nation ID**:

**PATIENT GENDER**:  
 Male       Female
   
**Patient Age**: 
  
**Patient phone**: 
  
**Patient PCR**:  
 Positive       Negative
   
**Patient Does**: 
  
**Address**:

ID	Name	Nation ID	Gender	Age	Phone	Address	pcr	Doctor	nurse	does
1	Ahmed	30107151401515	Male	21	01279827842	Alex	positive	1	2	Coronavac
2	Hossam	30107014045352	Male	30	1279827842	cairo	Negative	3	2	Oxford
3	khaled	30105191826374	Male	33	1004651157	Alex	Negative	2	1	Oxford
4	Rawan	30107986345287	Female	26	01001918125	Benha	Reserve	3	1	Sputnik
5	Ahmed	30107151401515	Male	21	01279827842	Alex	positive	3	2	Coronavac
6	Ahmed	30107151401515	Male	21	01279827842	Alex	positive	1	2	Coronavac
7	khaled	30105689732145	Male	19	01202694321	Benha	positive	1	2	Oxford
8	Ahmed	30107151401515	Male	21	01279827842	Alex	positive	1	2	Coronavac
9	Ahmed	30107151401515	Male	30	1279827842	cairo	positive	3	2	Moderna

**Update**      **Delete**      **Back**



**PATIENT NAME**

**NATION ID**

**PATIENT GENDER**

**Patient Age**

**Patient phone**

**Patient PCR**

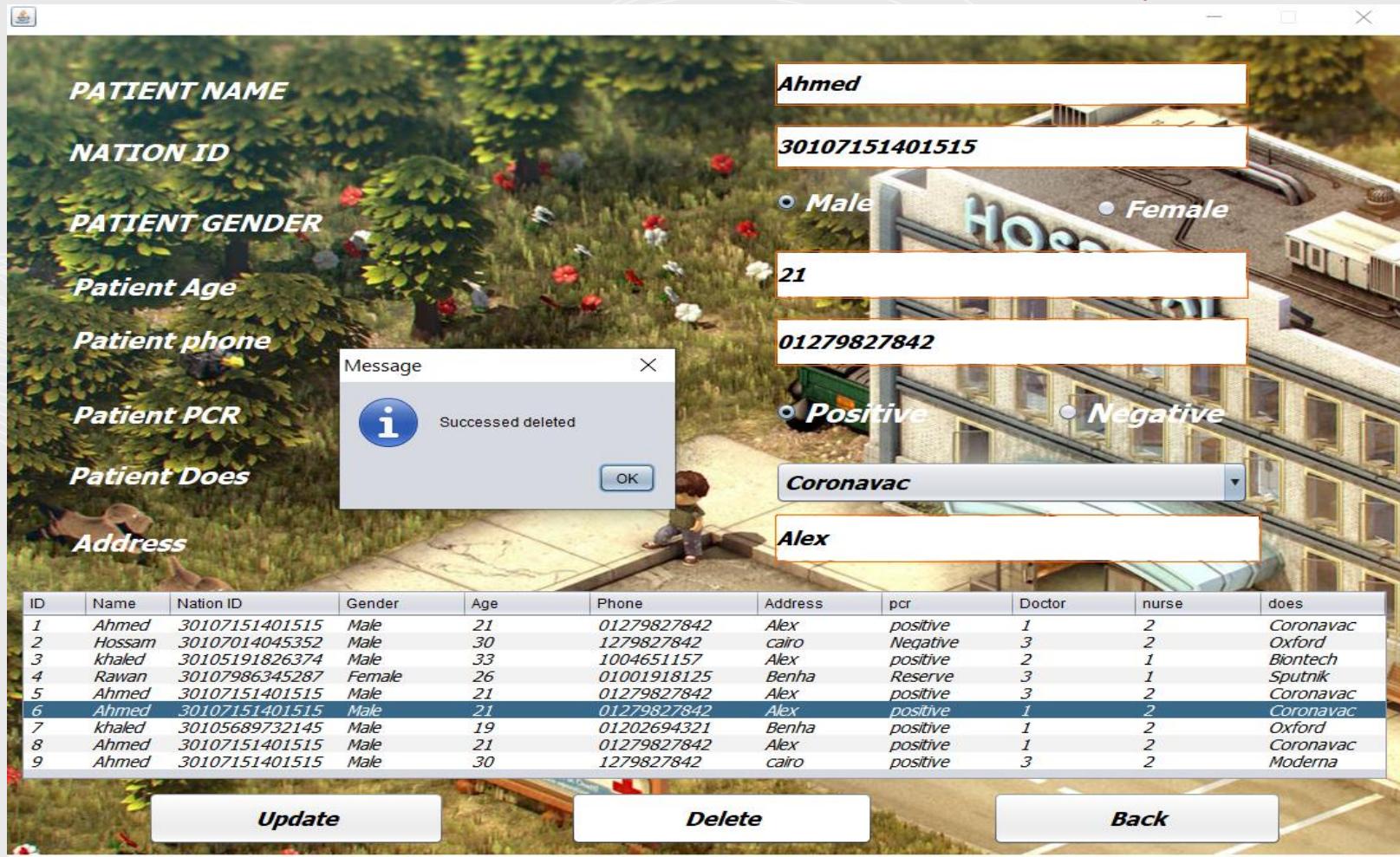
**Patient Does**

**Address**

**ID**    **Name**    **Nation ID**    **Gender**    **Age**    **Phone**    **Address**    **pcr**    **Doctor**    **nurse**    **does**

1	Ahmed	30107151401515	Male	21	01279827842	Alex	positive	1	2	Coronavac
2	Hossam	30107014045352	Male	30	1279827842	cairo	Negative	3	2	Oxford
3	khaled	30105191826374	Male	33	1004651157	Alex	Negative	2	1	Oxford
4	Rawan	30107986345287	Female	26	01001918125	Benza	Reserve	3	1	Sputnik
5	Ahmed	30107151401515	Male	21	01279827842	Alex	positive	3	2	Coronavac
6	Ahmed	30107151401515	Male	21	01279827842	Alex	positive	1	2	Coronavac
7	khaled	30105689732145	Male	19	01202694321	Benza	positive	1	2	Oxford
8	Ahmed	30107151401515	Male	21	01279827842	Alex	positive	1	2	Coronavac
9	Ahmed	30107151401515	Male	30	1279827842	cairo	positive	3	2	Moderna

**Update**    **Delete**    **Back**











# Database



Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

**SCHEMAS**

Filter objects

- campany
- clinic**
  - Tables
    - doctor
    - Columns
      - doc\_id
      - doc\_name
      - doc\_nation
      - doc\_spec
      - doc\_gender
      - doc\_age
      - doc\_phone
      - doc\_address
    - Indexes
    - Foreign Keys
    - Triggers
  - does
  - nurse

Administration Schemas

Information

Schema: clinic

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

```

1   #create database clinic
2
3   #create table does(does_id int primary key auto_increment,doestype varchar(50),count int);
4   #insert into does (doestype,count) values ("Oxford",0),("Biontech",0),("Johnson",0),("Sputnik",0),("Sinopharm",0),("Coronavac",0),("Moderna",0);
5
6   #create table doctor(doc_id int primary key auto_increment,doc_name varchar(50),doc_nation varchar(14),doc_spec varchar(50),doc_gender varchar(10),doc_age int,doc_ph
7   #insert into doctor (doc_name,doc_nation,doc_spec,doc_gender,doc_age,doc_phone,doc_address) values ("Ahmed",30107151401515,"Pediatrician","Male",30,01279827842,"Ben
8   #("Taha",30105191826374,"Endocrinologist","Male",32,01004651157,"Alex"),("Rawan",30107986345287,"Neurologist","Female",26,01001918125,"Benha"),
9   #("khaled",30107151401515,"Pediatrician","Male",30,01279827842,"tanta"),("omar",30105191826374,"Endocrinologist","Male",32,01004651157,"monfia"),
10  #("Heba",30107986345287,"Neurologist","Female",26,01001918125,"Assuit");
11
12  #create table nurse(nurse_id int primary key auto_increment,nurse_name varchar(50),nurrsse_nation varchar(14),nurse_gender varchar(10),nurse_age int,nurse_phone var
13  #insert into nurse (nurse_name,nurrsse_nation,nurse_gender,nurse_age,nurse_phone,nurse_address) values ("omar",30107151401515,"Male",30,01279827842,"cairo"),
14  #("khaled",30105191826374,"Male",32,01004651157,"Alex"),("Rawan",30107986345287,"Female",26,01001918125,"Benha"),("Fatma",30107986345287,"Female",26,010019181
15
16  #create table patient(pat_id int primary key auto_increment,pat_name varchar(50),pat_nation varchar(14),pat_gender varchar(10),pat_age int,pat_phone varchar(11),pat_
17
18  #insert into patient (pat_name,pat_nation,pat_gender,pat_age,pat_phone,pat_address,pcr,doc_id,nurse_id,does_id) values ("Hessen",30107151401515,"Male",30,0127982784
19  #("khaled",30105191826374,"Male",32,01004651157,"Alex",null,2,1,3),("Rawan",30107986345287,"Female",26,01001918125,"Benha",null,3,1,4),("Fatma",30107986345287
20
21  • select * from doctor;
  
```

doctor 39 ×

doc_id	doc_name	doc_nation	doc_spec	doc_gender	doc_age	doc_phone	doc_address
1	Ahmed	30107151401515	Pediatrician	Male	30	1279827842	Benha
2	Taha	30105191826374	Endocrinologist	Male	32	1004651157	Alex
3	Rawan	30107986345287	Neurologist	Female	26	1001918125	Benha
4	khaled	30107151401515	Pediatrician	Male	35	1279827842	tanta
7	omnia	12345678912345	dentist	Femal	20	01012345678	benha

Result Grid Apply Revert



## Navigator

## SCHEMAS

- Filter objects
- ◆ doc\_age
- ◆ doc\_phone
- ◆ doc\_address
- ▶ Indexes
- ▶ Foreign Keys
- ▶ Triggers
- ▶ does
- ▶ nurse
  - Columns
  - ◆ nurse\_id
  - ◆ nurse\_name
  - ◆ nursse\_nation
  - ◆ nurse\_gender
  - ◆ nurse\_age
  - ◆ nurse\_phone
  - ◆ nurse\_address
- ▶ Indexes
- ▶ Foreign Keys
- ▶ Triggers
- ▶ patient
  - Columns
  - ▶ Indexes
  - ▶ Foreign Keys

## clinic\*

```
17  
18      #insert into patient (pat_name,pat_nation,pat_gender,pat_age,pat_phone,pat_address,pcr,doc_id,nurse_id,does_id) values ("Hessen",30107151401515,"Male",30,0127982784,  
19      #("khaled","30105191826374","Male",32,01004651157,"Alex",null,2,1,3),("Rawan","30107986345287","Female",26,01001918125,"Benha",null,3,1,4),("Fatma","30107986345287").  
20  
21 •   select * from does;
```

## Result Grid

does_id	doestype	count
1	Oxford	0
2	Biontech	0
3	Johnson	0
4	Sputnik	0
5	Sinopharm	0
6	Coronavac	0
7	Moderna	0
*	HULL	HULL



## Administration Schemas

## Information

Schema: clinic

## does 45 \*

Apply Revert



Navigator

SCHEMAS

Filter objects

- ◆ doc\_age
- ◆ doc\_phone
- ◆ doc\_address

- ▶ Indexes
- ▶ Foreign Keys
- ▶ Triggers

▶ does

▶ nurse

- ▶ Columns
  - ◆ nurse\_id
  - ◆ nurse\_name
  - ◆ nurse\_nation
  - ◆ nurse\_gender
  - ◆ nurse\_age
  - ◆ nurse\_phone
  - ◆ nurse\_address

- ▶ Indexes
- ▶ Foreign Keys
- ▶ Triggers

▶ patient

- ▶ Columns
- ▶ Indexes
- ▶ Foreign Keys

Administration

Schemas

Information

Schema: clinic

clinic\*



Limit to 1000 rows



```
17  
18  #insert into patient (pat_name,pat_nation,pat_gender,pat_age,pat_phone,pat_address,pcr,doc_id,nurse_id,does_id) values ("Hessen",30107151401515,"Male",30,0127982784,  
19  #("khaled","30105191826374","Male",32,01004651157,"Alex",null,2,1,3),("Rawan","30107986345287","Female",26,01001918125,"Benha",null,3,1,4),("Fatma","30107986345287").  
20  
21 • select * from nurse;
```

Result Grid

Filter Rows:

Edits



Wrap Cell Content:

nurse_id	nurse_name	nurse_nation	nurse_gender	nurse_age	nurse_phone	nurse_address
1	omar	30107151401515	Male	30	01279827842	cairo
3	Rawan	30107986345287	Femal	26	01001918125	Benha
4	Fatma	30107986345287	Femal	26	01001918125	monfia
5	Rana	30107151401515	Femal	26	01279827842	shobra
6	heba	30107150151415	Femal	30	01279527862	cairo
8	Heba	3010715140151	Male	30	01279827842	Alex
*	NULL	NULL	NULL	NULL	NULL	NULL



nurse 44

Apply

Revert

**Navigator**

**Schema: clinic\***

**Schemas**

Filter objects

- doc\_age
- doc\_phone
- doc\_address
- Indexes
- Foreign Keys
- Triggers
- does
- nurse

  - Columns
    - nurse\_id
    - nurse\_name
    - nurse\_nation
    - nurse\_gender
    - nurse\_age
    - nurse\_phone
    - nurse\_address
  - Indexes
  - Foreign Keys
  - Triggers

- patient

  - Columns
  - Indexes
  - Foreign Keys

**Administration**

**Information**

**Result Grid**

Limit to 1000 rows

```

17
18      #insert into patient (pat_name,pat_nation,pat_gender,pat_age,pat_phone,pat_address,pcr,doc_id,nurse_id,does_id) values ("Hessen",30107151401515,"Male",30,0127982784;
19      #("khaled","30105191826374","Male",32,01004651157,"Alex",null,2,1,3),("Rawan","30107986345287","Female",26,01001918125,"Benha",null,3,1,4),("Fatma","30107986345287";
20
21 • select * from patients;
    
```

Filter Rows:

Edit: Export/Import: Wrap Cell Content:

pat_id	pat_name	pat_nation	pat_gender	pat_age	pat_phone	pat_address	pcr	doc_id	nurse_id	does_id
1	Ahmed	30107151401515	Male	21	01279827842	Alex	positive	1	2	6
2	Hossam	3010714045352	Male	30	1279827842	cairo	positive	3	2	2
3	khaled	30105191826374	Male	33	1004651157	Alex	positive	2	1	2
4	Rawan	30107986345287	Female	26	01001918125	Benha	Reserve	3	1	4
5	Ahmed	30107151401515	Male	21	01279827842	Alex	Negative	3	2	7
6	Ahmed	30107151401515	Male	21	01279827842	Alex	positive	1	2	6
7	khaled	30105689732145	Male	19	01202694321	Berha	positive	1	2	1
8	Ahmed	30107151401515	Male	21	01279827842	Alex	positive	1	2	6
9	Ahmed	30107151401515	Male	30	1279827842	cairo	positive	3	2	7
10	mayada	12345678910110	Femal	21	01093553691	Monifiya	NULL	1	2	NULL
11	mayada	12345678910110	Femal	21	01093553691	Monifiya	NULL	1	2	NULL
12	ahmed	11223344556677	Male	20	01093553691	benha	NULL	1	2	NULL
13	Yasser	30107151401414	Male	20	01279827842	Alex	NULL	1	2	NULL
14	mayada	00000000000000	Female	20	01093553691	Albajour	Reserve	1	2	NULL
15	omar	12345678912345	Male	123	01010331410	benha	Reserve	1	2	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

patient 43 x Apply Revert

# **GUI and Database Connection code**

```
/*
public class show_all_doctor extends javax.swing.JFrame {

    /**
     * Creates new form show_all_doctor
     */
    DefaultTableModel dm;
    Connection con;
    ArrayList<Integer> doc_ids = new ArrayList<>();

    public show_all_doctor() {
        initComponents();
        this.setLocationRelativeTo(null);
        dm = new DefaultTableModel();
        dm.addColumn("ID");
        dm.addColumn("Name");
        dm.addColumn("Nation ID");
        dm.addColumn("Specialization");
        dm.addColumn("Gender");
        dm.addColumn("Age");
        dm.addColumn("Phone");
        dm.addColumn("Address");

        try {
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/clinic", "root", "ahmedhosny12345");
        } catch (SQLException ex) {
            Logger.getLogger(show_all_doctor.class.getName()).log(Level.SEVERE, null, ex);
        }
        filldata();
    }

    private void filldata() {
        dm.setRowCount(0);
        Statement st;
        ResultSet rs;
        String query = "SELECT * FROM doctor";
        try {
            st = con.createStatement();
            rs = st.executeQuery(query);
            while (rs.next()) {
                int id = rs.getInt("id");
                String name = rs.getString("name");
                String nation_id = rs.getString("nation_id");
                String specialization = rs.getString("specialization");
                String gender = rs.getString("gender");
                int age = rs.getInt("age");
                String phone = rs.getString("phone");
                String address = rs.getString("address");
                doc_ids.add(id);
                dm.addRow(new Object[]{id, name, nation_id, specialization, gender, age, phone, address});
            }
        } catch (SQLException ex) {
            Logger.getLogger(show_all_doctor.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    private void initComponents() {
        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jTable1.setModel(dm);
        jScrollPane1.setViewportView(jTable1);

        jButton1.setText("Back");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setText("Logout");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jButton1)
                .addGap(18, 18, 18)
                .addComponent(jButton2)
                .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jButton1)
                    .addComponent(jButton2))
                .addContainerGap())
        );

        pack();
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        new login().setVisible(true);
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        new login().setVisible(true);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // </pre>
    // <code>
    // </code>
    // <pre>
    // </pre>
```

```
private void filldata(){  
    try {  
        dm.setRowCount(0);  
        PreparedStatement stm = con.prepareStatement("select * from doctor");  
        ResultSet res = stm.executeQuery();  
        while(res.next()) {  
            doc_ids.add(res.getInt(1));  
            String[] data = {res.getString(1),res.getString(2), res.getString(3), res.getString(4), res.getString(5),res.getString(6),res.getString(7),res.getString(8)};  
            dm.addRow(data);  
        }  
        show_data.setModel(dm);  
    } catch (SQLException ex) {  
        JOptionPane.showMessageDialog(this,"erorr");  
        Logger.getLogger(show_all_doctor.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

```
private void show_dataMouseClicked(java.awt.event.MouseEvent evt) {  
  
    doc_name.setText(show_data.getValueAt(show_data.getSelectedRow(), 1).toString());  
    doc_id.setText(show_data.getValueAt(show_data.getSelectedRow(), 2).toString());  
  
    String gender = show_data.getValueAt(show_data.getSelectedRow(), 4).toString();  
    if ("Male".equals(gender)) {  
        rbtn_male.setSelected(true);  
    }  
    else {  
        rbtn_femal.setSelected(true);  
    }  
  
    age.setText(show_data.getValueAt(show_data.getSelectedRow(), 5).toString());  
    spes.setText(show_data.getValueAt(show_data.getSelectedRow(), 3).toString());  
    phone.setText(show_data.getValueAt(show_data.getSelectedRow(), 6).toString());  
    address.setText(show_data.getValueAt(show_data.getSelectedRow(), 7).toString());  
}  
//
```

```
private void doc_deleteActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try {  
        // TODO add your handling code here  
  
        PreparedStatement stmt = con.prepareStatement("delete from doctor where doc_id=?");  
        stmt.setInt(1, doc_ids.show_data.getSelectedRow());  
        int res = stmt.executeUpdate();  
        if (res > 0) {  
            JOptionPane.showMessageDialog(this, "Successed deleted");  
            filldata();  
        } else {  
            JOptionPane.showMessageDialog(this, "Faild deleted");  
        }  
  
    } catch (SQLException ex) {  
        Logger.getLogger(show_all_doctor.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try {  
        String name = this.name.getText();  
        String nationn = id.getText();  
        String gender;  
        if (rbtn_male.isSelected()) {  
            gender = "Male";  
        } else {  
            gender = "Femal";  
        }  
        int age = Integer.parseInt(this.age.getText());  
        String addr = address.getText();  
        String pho = phone.getText();  
  
        PreparedStatement stnt = con.prepareStatement("insert into patient (pat_name,pat_nation,pat_gender,pat_age,pat_phone,pat_address,pcr,doc_id,nurse_id,does_id)values(?,?,?,?,?,?,?,?,?,?)");  
        stnt.setString(1, name);  
        stnt.setString(2, nationn);  
        stnt.setString(3, gender);  
        stnt.setInt(4, age);  
        stnt.setString(5, pho);  
        stnt.setString(6, addr);  
        stnt.setString(7, null);  
        stnt.setInt(8, 1);  
        stnt.setInt(9, 2);  
        stnt.setString(10, null);  
        int res = stnt.executeUpdate();  
        if (res > 0) {  
            JOptionPane.showMessageDialog(this, "Account Create Successfull");  
        } else {  
            JOptionPane.showMessageDialog(this, "Faild Created");  
        }  
    } catch (SQLException ex) {  
        Logger.getLogger(AddDoctor.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        String name = this.name.getText();  
        String id = this.id.getText();  
        String gender;  
        if (rbtn_male.isSelected()) {  
            gender = "Male";  
        } else {  
            gender = "Female";  
        }  
        int age = Integer.parseInt(this.age.getText());  
        String pho = phone.getText();  
        String add = address.getText();  
        PreparedStatement stmt = con.prepareStatement("update patient set pat_name=? ,pat_nation=? ,pat_gender=? ,pat_age=? ,pat_phone=? ,pat_address=? where pat_nation='"+id+"'" );  
        stmt.setString(1, name);  
        stmt.setString(2, id);  
        stmt.setString(3, gender);  
        stmt.setInt(4, age);  
        stmt.setString(5, pho);  
        stmt.setString(6, add);  
        int res = stmt.executeUpdate();  
        if (res > 0) {  
            JOptionPane.showMessageDialog(this, "Success update");  
        } else {  
            JOptionPane.showMessageDialog(this, "Faild update");  
        }  
    } catch (SQLException ex) {  
        Logger.getLogger(show_all_patient.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

# 04.

## CONCLUSION

COVID-19 Management System reviews cases on many aspects of healthcare in residential settings. By providing a range of services that help with that. The increasing numbers of elderly people, the increase in the spread of the disease, and the improvement of survival rates for people suffering from those conditions and diseases. Medical Tech recommends to improve health care at home. Covering healthcare technologies, providing a range of doctors' advice.