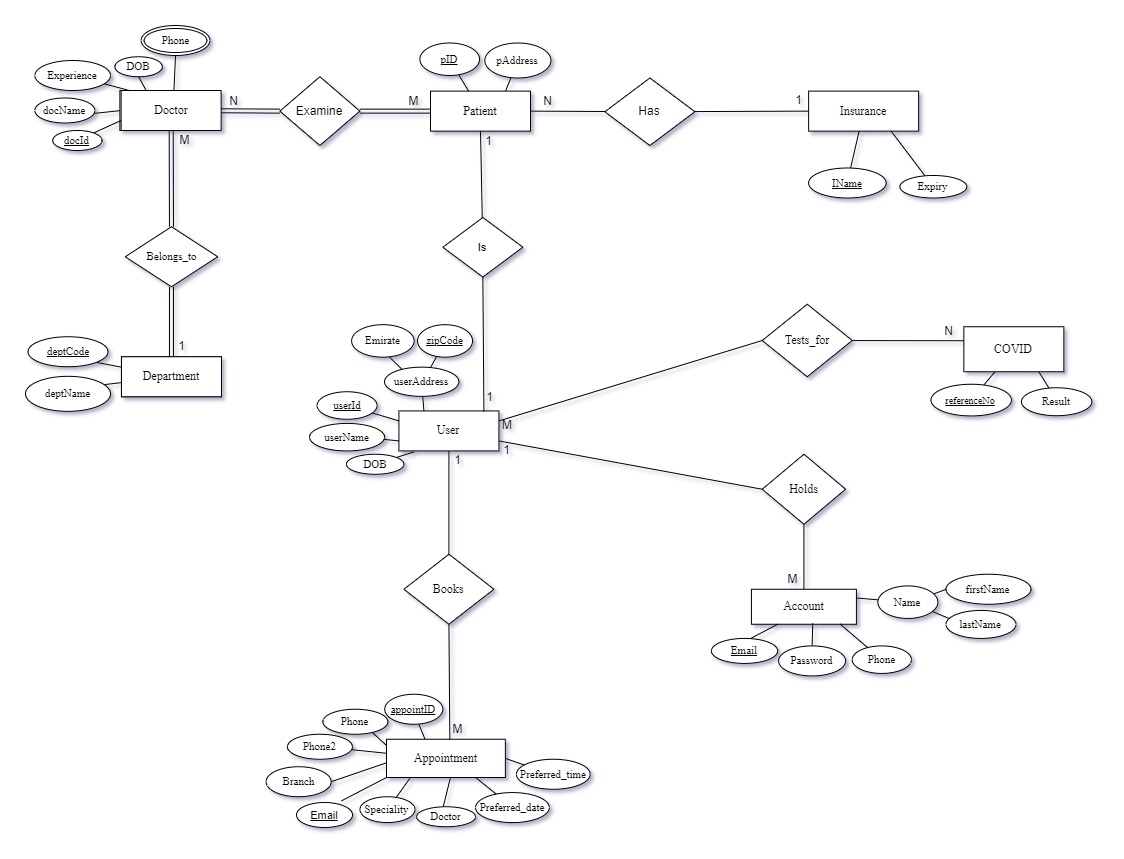
**HEALIFE HOSPITAL**

**PROJECT REPORT**

1. **Brief Description of Implementation**

ini\_set("SMTP","localhost");ini\_set("smtp\_port",25);ini\_set("sendmail\_from","turemitente@gmail.com"); $subject = "TEST" ;$message = "User message" ;$user\_email = "xxxxxxxxxxx@gmail.com" ; // valid POST email address $headers = "From: $user\_email " ;$headers .= "Reply-To: $too " ;$headers .= "Return-Path: $too " ;$headers .= "X-Mailer: PHP/" . phpversion (). " " ;$headers .= 'MIME-Version: 1.0' . " " ;$headers .= 'Content-type: text/html; UTF-8' . " " ; if( mail ( $too , $subject , $message , $headers )) echo 'SENT' ;

1. **Database Conceptual Schema**

****

**Assumptions**

1. `docID` is the **Primary Key** of the **Doctor** entity.

2. `Phone` attribute of **Doctor** is multivalued.

3. `deptCode` is the **Primary Key** of the **Department** entity.

4. `userID` is the **Primary Key** of the **User** entity.

5. `userAddress` is a composite attribute in the **User**.

6. `pID` is the **Primary Key** of the **Patient** entity.

7. Since a user is not necessarily a patient, **User** and **Patient** can have different addresses

8. `IName` is the **Primary Key** of the **Insurance** entity, because there cannot be two insurance companies with the same name.

9. `referenceNo` is the **Primary Key** of the **COVID** entity.

10. `Email` is the **Primary Key** of the **Account** entity.

11. `Name` is a composite attribute in the **Account**.

12. `appointmentID` and `Email` are the **Primary Keys** of the **Appointment** entity.

**Constraints**

1. A **Department** must have one or more doctors.

2. A **Doctor** must belong to one department.

3. **Doctor** must examine one or more patients.

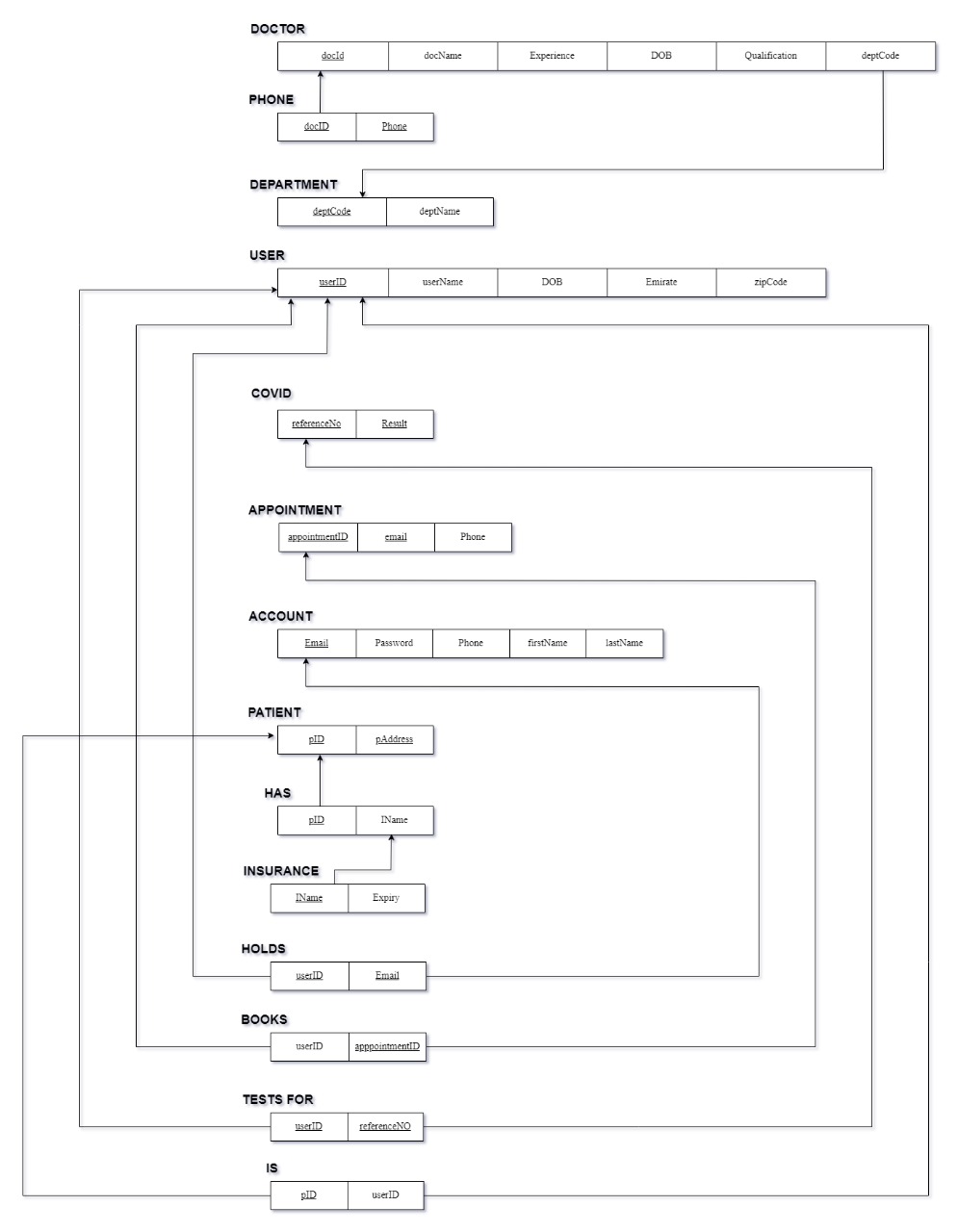
4. A **Patient** must be examined by one or more doctors.

5. A **Patient** can have only one Insurance but the **Insurance** can have many patients.

6. **User** can test for **COVID** multiple times.

7. A **User** can have multiple accounts but one **Account** is handled by only one.

1. **Relational Model**

****

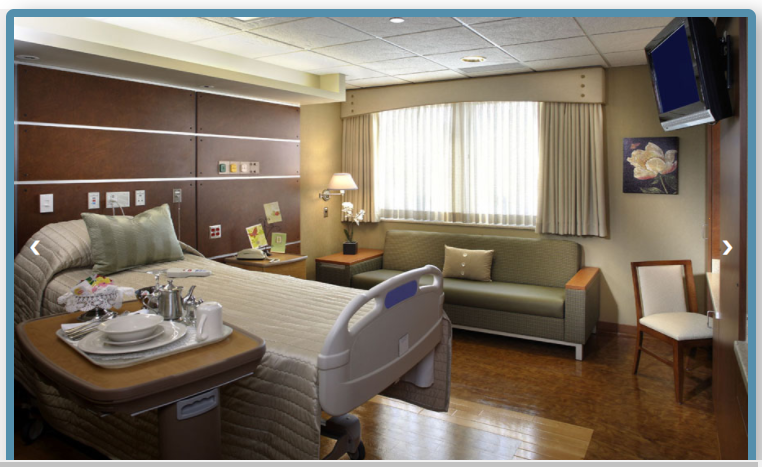
1. **User Guide**

Our team has decided to explore, design, and code and service-oriented platform, therefore creating ‘Healife Hospital’. Our database is based on a hospital along with related entities.

1. The index.html file displays the code of our ‘Home’ page. The home page welcomes you to our main concept, logo, and slogan. The user will notice a warm selection of colors and a simple layout to welcome the users and make navigation easy. Followed by some statistics to display our success so far. We have a widespread map with branches in Ajman, Fujairah, Ras al Khaimah, Sharjah, and Dubai.

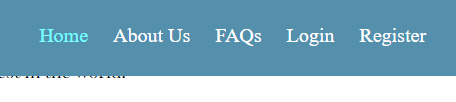


1. The hospital offers services and special facilities for its patients like Suites and VIP rooms. The hospital environment is made comfortable and rejuvenating giving a sense of hope and healing to its visitors and patients.

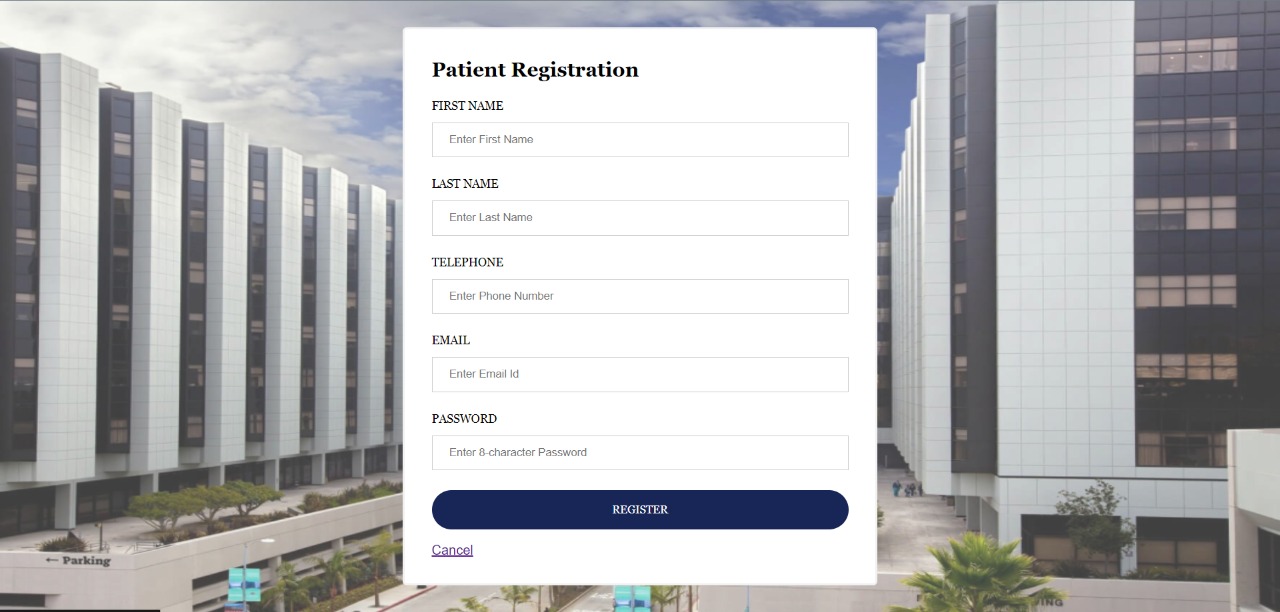


1. In the navigation bar, the user will notice different linked web pages. The user can proceed to ‘ABOUT US’ one can read and understand the technicalities of the hospital. We can check out some new facilities added to our hospitals and have a look at our doctors.

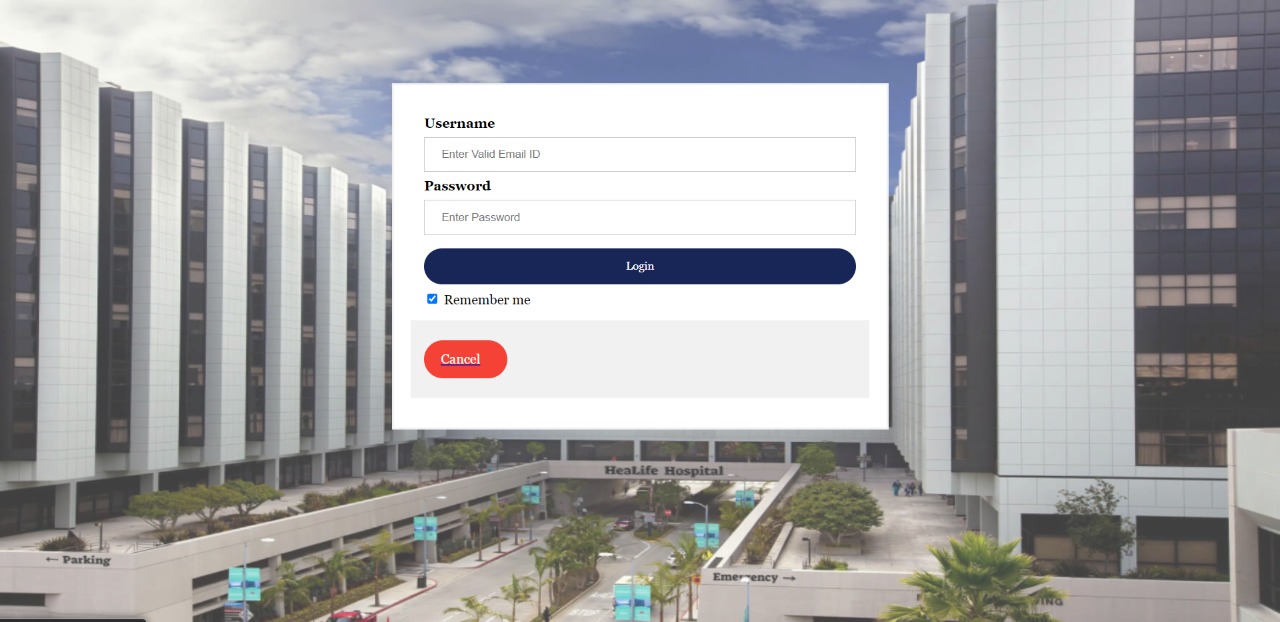




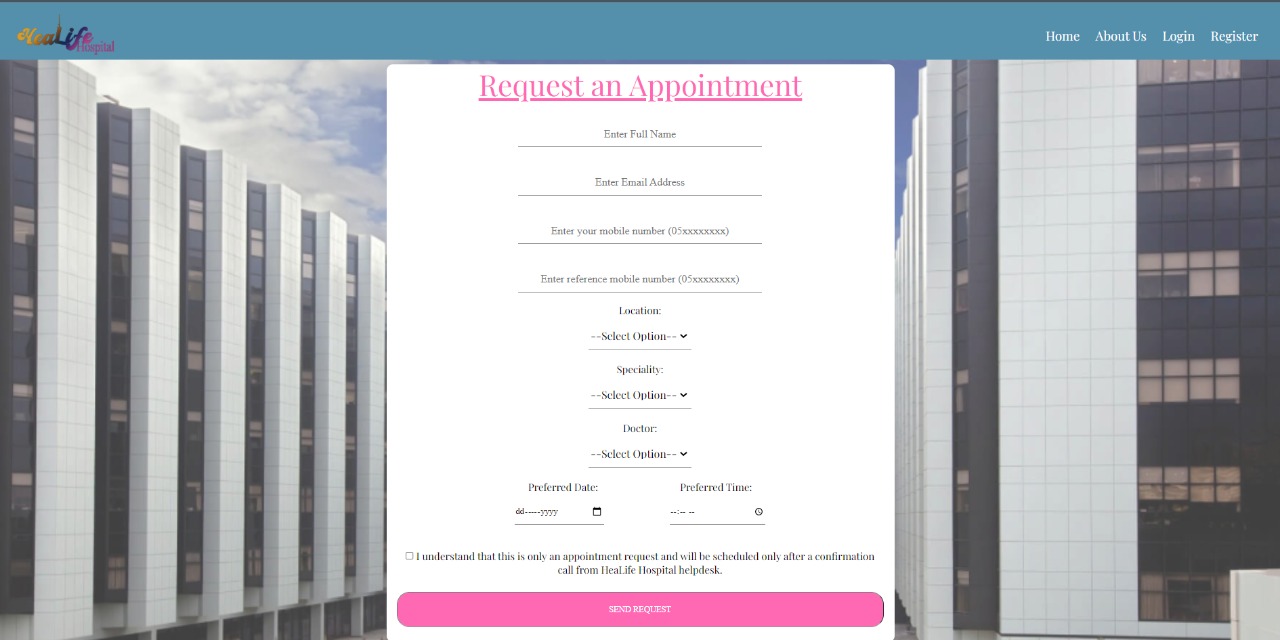
1. The Register Page allows the first-time User to enter their First Name, Last Name, Telephone number, Email id, Password. This Register is connected to our database table called ‘Account’, which has attributes such as Email, Password, Phone First name, and Last name. When the user clicks on the register button, our webpage connects to PHP of the Xampp server, which is then connected to our database in MySQL. Only once the user registers themselves they can move on to book an appointment.



1. If the user revisits the website, they can simply go to the ‘Login Page’ and enter their email ID and Password. This page is connected to the database via the PHP text, which retrieves the data from the account table and compares it with the details entered in the text fields. If the fields match, it is a success statement.



1. When user logins in or registers they are redirected to the appointment page. The user will have to enter their necessary details and they have to then send a request to get confirmed. When the user sends the request, all the details will be sent and added to the SQL database with the help of PHP where we add the script for the connection.



1. **How the work was split**

|  |  |  |
| --- | --- | --- |
| **SID** | **NAME** | **WORK** |
| **20000932** | Majid Alamoudi | * Xampp * MySql Database – Creating tables, attributes and inserting values * Report – Database conceptual schema, relational model, * Registration webpage – HTML, CSS * FAQs webpage – HTML, CSS * PHP – Registration * ER Diagram |
| **20000944** | Vanessa Fernando | * Relational model * About Us Webpage – HTML, CSS * Login Webpage – HTML, CSS * PHP – Login * Report – User Guide |
| **20000918** | Zobia Tayyab | * Relational model * Compilation of website * Research on sending an email on submission appointment form * Research on XAMPP server * Fixing errors in MySql and PHP * Homepage – HTML, CSS * Appointment – HTML, CSS |