KJA CLINIC DATABASE

Course: ITE442

Instructor: Atikom Srivallop

Members:

Kimseng Houy 1809170001

Alexader Vieten 1903070008

Justin Himarua 1902150006

Table of content

1. Introduction \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ Page 3
2. Normalization\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ Page 4-9
3. ER Diagram \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ Page 10
4. UML \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ Page 11
5. Database View\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ Page 12
6. Database Trigger \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_Page 13
7. WEBPAGE Queries \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ Page 14-18
8. Our Demo \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_Page 19
9. Further work \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ Page 20

Introduction

In this project, we have normalized KJA Clinic database from 1NF to 3NF and created website application to manage the appointment online, view the patient or add the patient, etc.

How to use our application?

It starts from

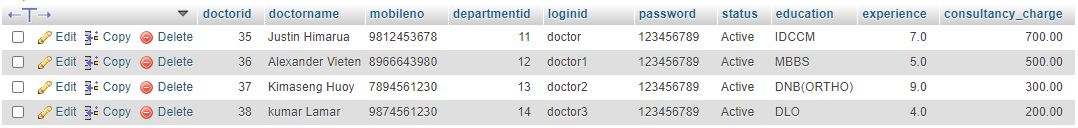
**Normalization**

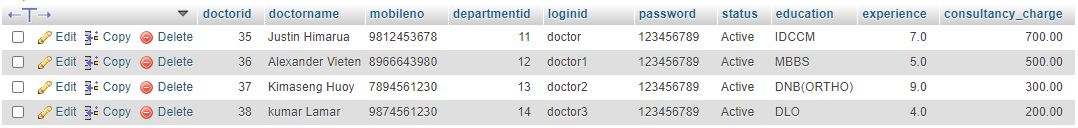
Normalization is a database plan method that diminishes information repetition and dispenses with unwanted qualities like Insertion, Update and Deletion Anomalies. Standardization rules separates bigger tables into more modest tables and connects them utilizing connections. The reason for Normalization in SQL is to kill excess (redundant) information and guarantee information is put away legitimately.

Normalization-1NF

When looking at our database we have a multitude of entities that follow the 1st NF but a few of these tables need to be slightly adjusted to make it complete such as for the doctor entity, our database will have to account for multi value inserts

**Doctor**

* Original Data with multi value Warranty
* After Normalization



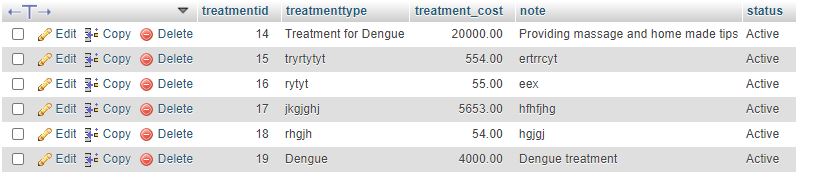
**Patient**

* After Normalization



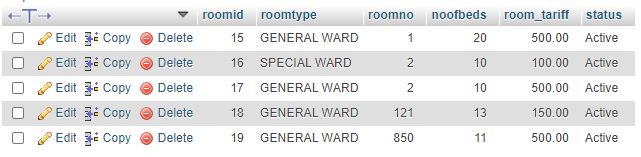
**Treatment**

* After Normalization



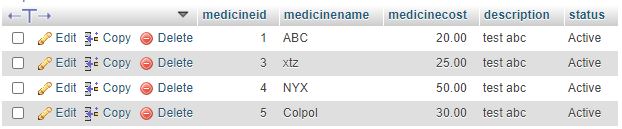
**Rooms**

* After Normalization



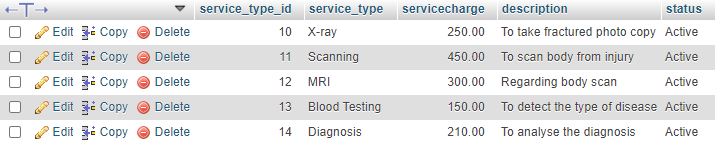
**Medicine**

* After Normalization

****

**Service\_type**

* After Normalization

****

**Department**

* After Normalization

****

**Admin**

* After Normalization

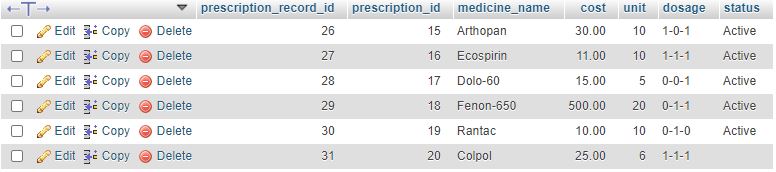
****

Normalization-2NF

dependent on the idea of full practical reliance. Second Normal Form applies to relations with composite keys, that is, relations with an essential key made out of at least two credits. A connection with a solitary property essential key is naturally in any event 2NF. A connection that isn't in 2NF may experience the ill effects of the update abnormalities.

* 2NF (No non-prime attribute is dependent on the subset of any candidate key)

**Prescription\_records**



* After Normalization

We created a new table called treatment\_records, the same was implemented for billing\_records and prescription\_records.

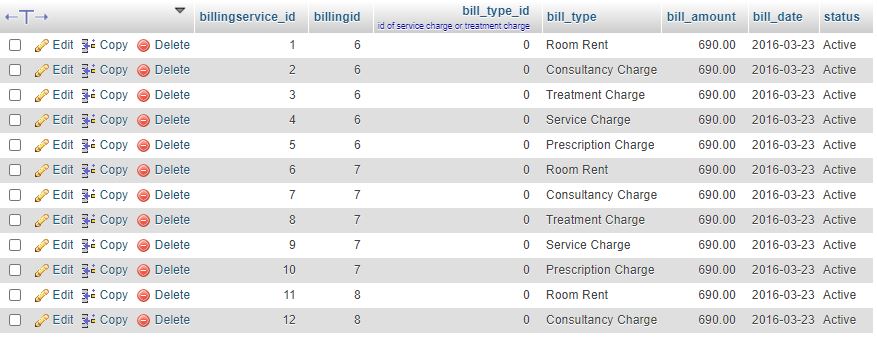
**Billing**

* After Normalization

****

**Billing\_Records**

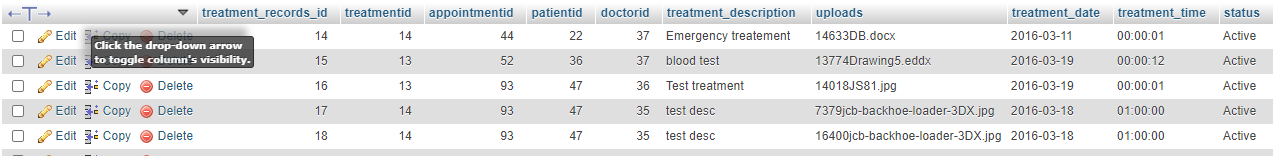
* After Normalization



**Treatment\_records**

**Treatment\_records**

* After Normalization

****

**Doctor Timings**

* After Normalization



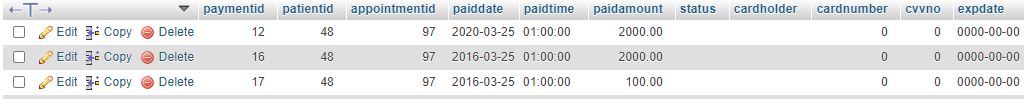
Normalization-3NF

A relation is in third typical structure, if there is no transitive reliance for non-prime credits just as it is in second ordinary structure.

**Normalization for Orders**



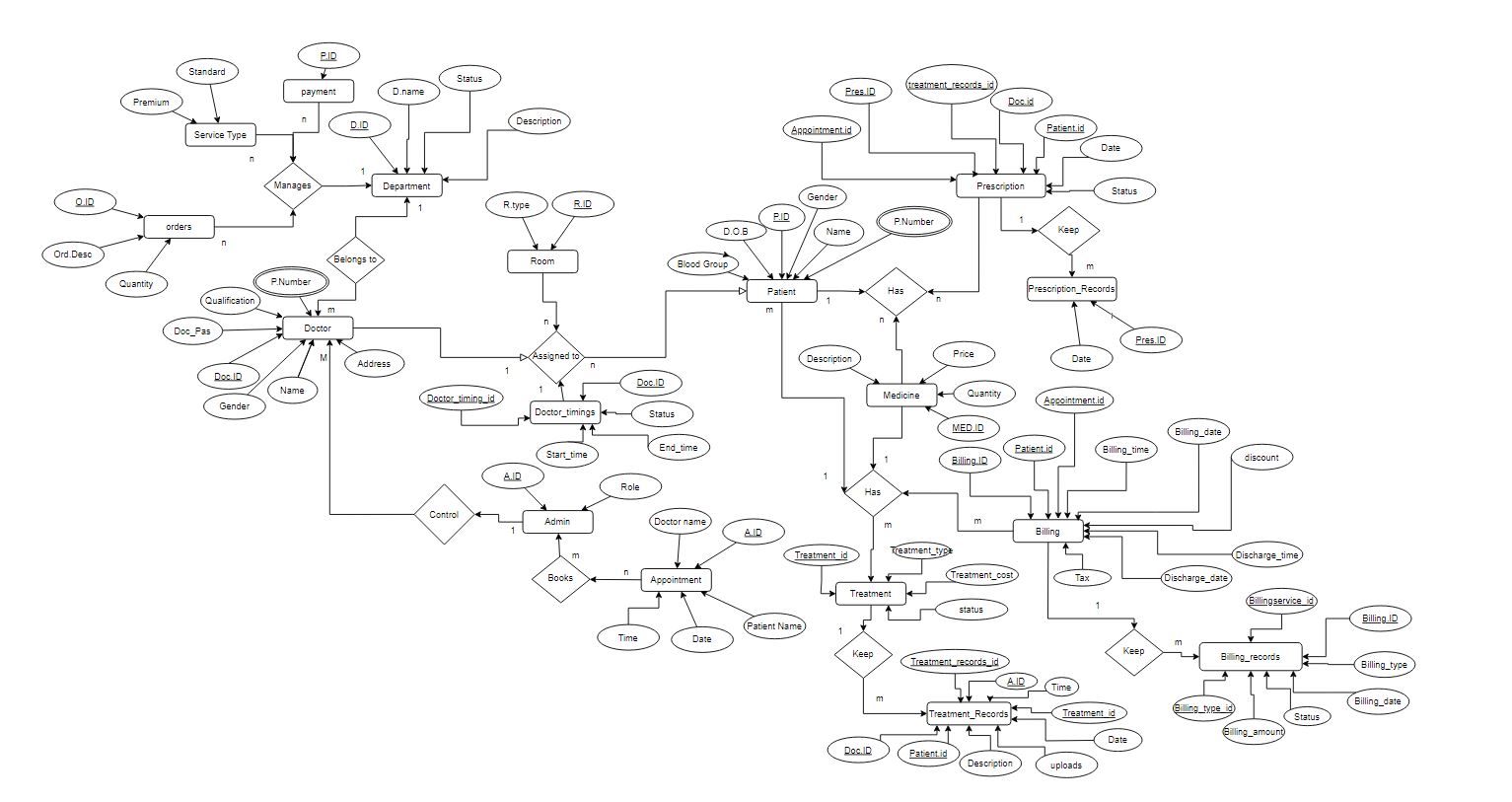
**Normalization for Payments**



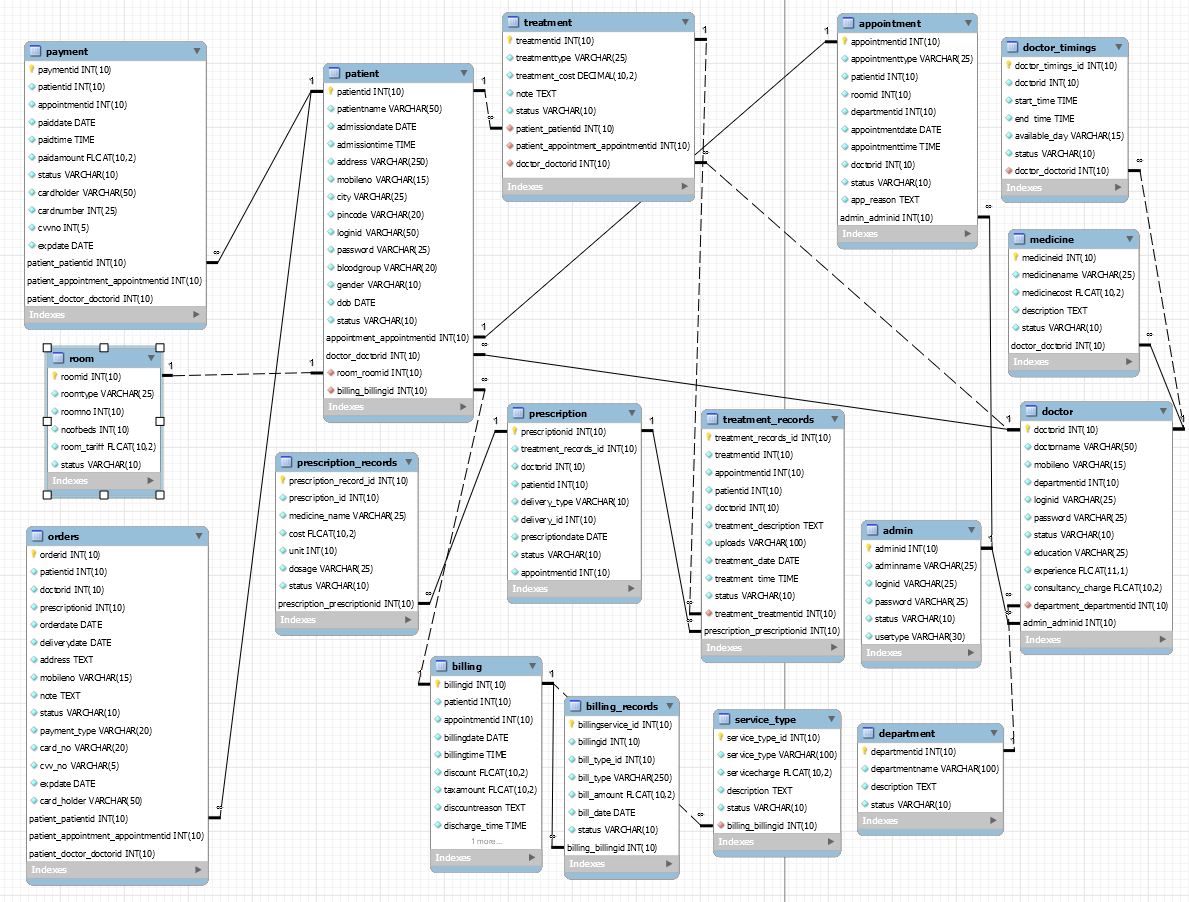
**Normalization for appointment**



ER DIAGRAM



UML



Database View

* View for Admin Patient Records:
* $sql ="SELECT \* FROM patient";
* $qsql = mysqli\_query($con,$sql);
* while($rs = mysqli\_fetch\_array($qsql))
* {
* echo "<tr>
* <td>$rs[patientname]<br>
* <strong>Login ID :</strong> $rs[loginid] </td>
* <td>
* <strong>Date</strong>: &nbsp;$rs[admissiondate]<br>
* <strong>Time</strong>: &nbsp;$rs[admissiontime]</td>
* <td>$rs[address]<br>$rs[city] -  &nbsp;$rs[pincode]<br>
* Mob No. - $rs[mobileno]</td>
* <td><strong>Blood group</strong> - $rs[bloodgroup]<br>
* <strong>Gender</strong> - &nbsp;$rs[gender]<br>
* <strong>DOB</strong> - &nbsp;$rs[dob]</td>
* <td align='center'>Status - $rs[status] <br>";
* if(isset($\_SESSION[adminid]))
* {
* echo "<a href='patient.php?editid=$rs[patientid]'>Edit</a> | <a href='viewpatient.php?delid=$rs[patientid]'>Delete</a> <hr>
* <a href='patientreport.php?patientid=$rs[patientid]'>View Report</a>";
* }
* echo "</td></tr>";
* }
* View for Patient Prescription Record:
* $sqlprescription\_records ="SELECT \* FROM prescription\_records LEFT JOIN medicine ON prescription\_records.medicine\_name=medicine.medicineid WHERE prescription\_records.prescription\_id='$rs[0]'";

Database Trigger

Trigger in patient table:

**Before insert:**

IF (NEW.pincode = 'CHAR') THEN

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Warning: patientid should be number or int!';

END IF;

**Explanation:** if patient user accidentally adds their pin code to CHAR or Letters it will show the Warning.

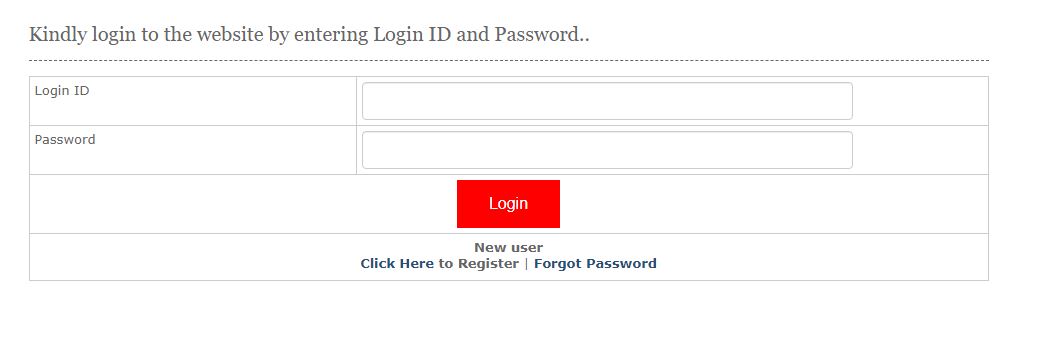
**After delete**

SELECT USER() INTO @userName;

SELECT NOW() INTO @deleteDatetime;

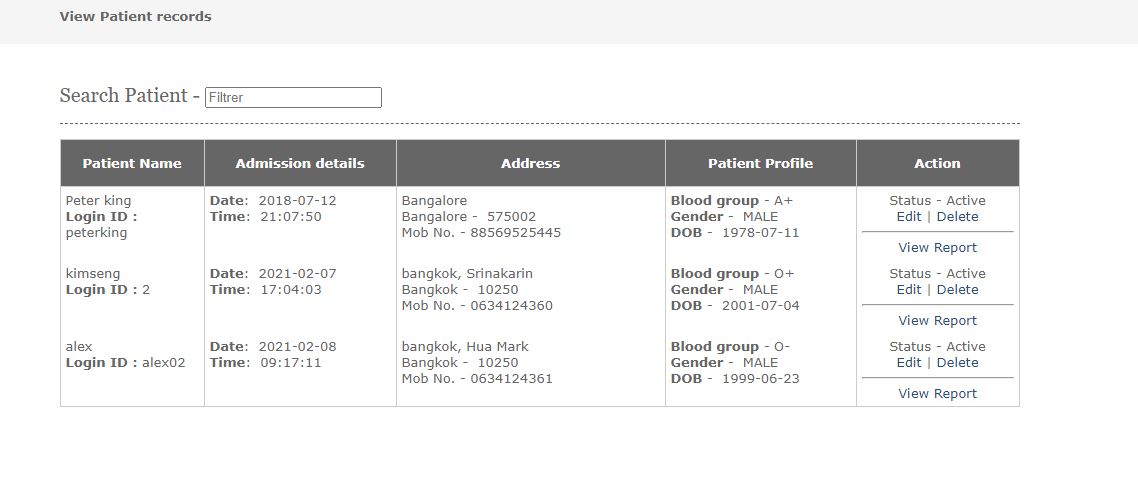
**Explanation:** when someone it deletes something in patient table it will displays who delete the records and what time is deleted.

WEBPAGE QUERIES - LOGIN





WEBPAGE QUERIES(View)



$sql ="SELECT \* FROM patient";

    $qsql = mysqli\_query($con,$sql);

    while($rs = mysqli\_fetch\_array($qsql))

    {

        echo "<tr>

          <td>$rs[patientname]<br>

      <strong>Login ID :</strong> $rs[loginid] </td>

          <td>

      <strong>Date</strong>: &nbsp;$rs[admissiondate]<br>

     <strong>Time</strong>: &nbsp;$rs[admissiontime]</td>

       <td>$rs[address]<br>$rs[city] -  &nbsp;$rs[pincode]<br>

Mob No. - $rs[mobileno]</td>

          <td><strong>Blood group</strong> - $rs[bloodgroup]<br>

<strong>Gender</strong> - &nbsp;$rs[gender]<br>

<strong>DOB</strong> - &nbsp;$rs[dob]</td>

          <td align='center'>Status - $rs[status] <br>";

if(isset($\_SESSION[adminid]))

{

      echo "<a href='patient.php?editid=$rs[patientid]'>Edit</a> | <a href='viewpatient.php?delid=$rs[patientid]'>Delete</a> <hr>

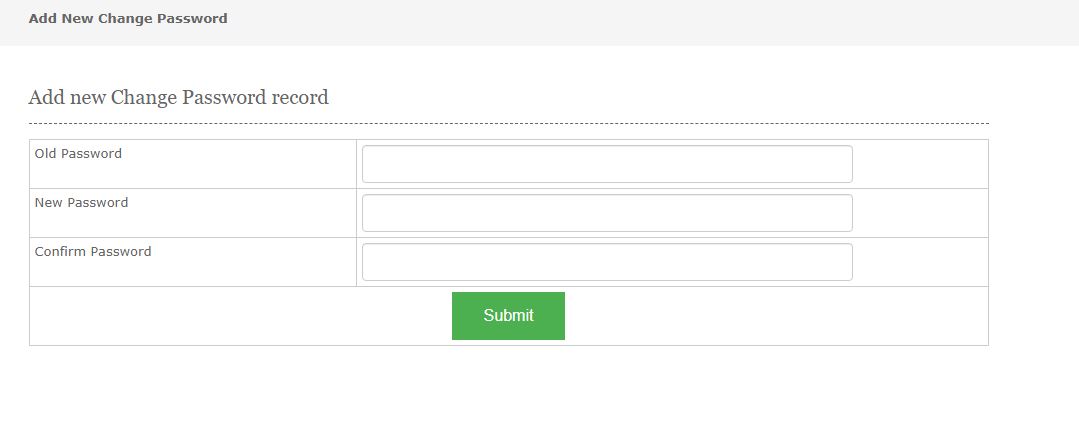
<a href='patientreport.php?patientid=$rs[patientid]'>View Report</a>";

}

      echo "</td></tr>";

    }

WEBPAGE QUERIES (UPDATE/Change)



if(isset($\_POST[submit]))

{

    $sql = "UPDATE patient SET password='$\_POST[newpassword]' WHERE password='$\_POST[oldpassword]' AND patientid='$\_SESSION[patientid]'";

    $qsql= mysqli\_query($con,$sql);

    if(mysqli\_affected\_rows($con) == 1)

    {

        echo "<script>alert('Password has been updated successfully..');</script>";

    }

    else

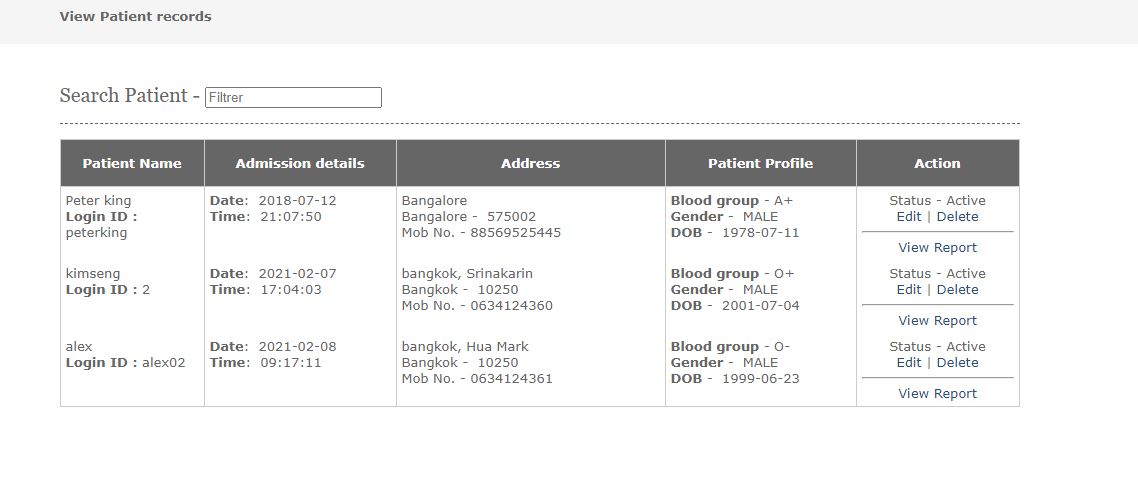
    {

        echo "<script>alert('Failed to update password..');</script>";

    }

}

WEBPAGE QUERIES (Delete)



if(isset($\_GET[delid]))

{

  $sql ="DELETE FROM patient WHERE patientid='$\_GET[delid]'";

  $qsql=mysqli\_query($con,$sql);

  if(mysqli\_affected\_rows($con) == 1)

  {

    echo "<script>alert('patient record deleted successfully..');</script>";

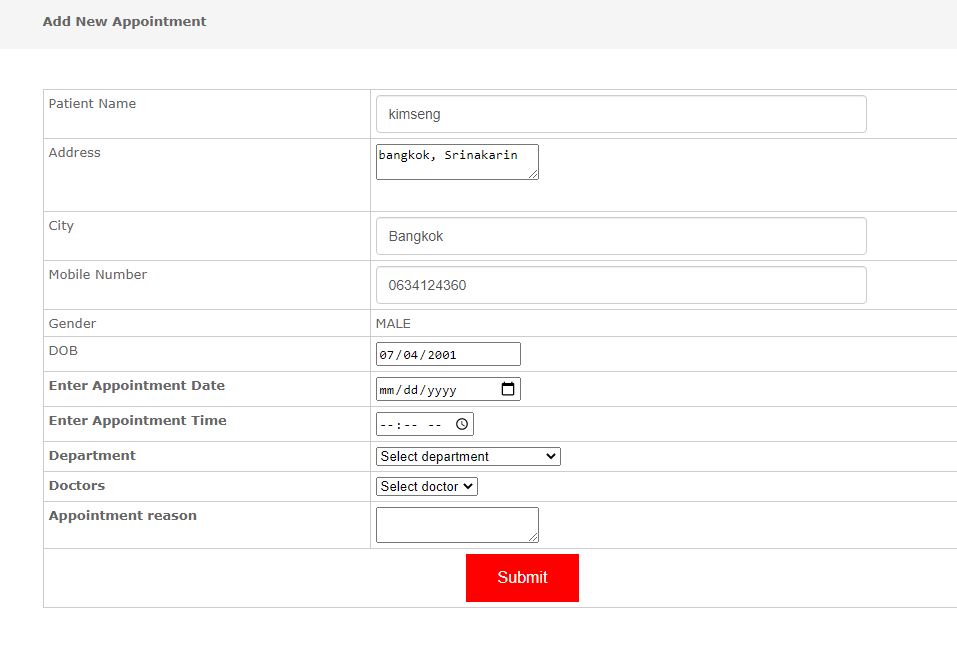
  }

}

**Other Queries for Delete:**

* Delete Treatment Record:
* $sql ="DELETE FROM treatment\_records WHERE appointmentid='$\_GET[delid]'";
* Delete Medicine:
* $sql ="DELETE FROM medicine WHERE medicineid='$\_GET[delid]'";
* Delete Doctor:
* $sql ="DELETE FROM doctor WHERE doctorid='$\_GET[delid]'";

Webpage Queries (Insert)



WEBPAGE QUERIES (Search)

WEB Application

$sql ="INSERT INTO appointment(appointmenttype,patientid,appointmentdate,appointmenttime,app\_reason,status,departmentid,doctorid) values('ONLINE','$lastinsid','$\_POST[appointmentdate]','$\_POST[appointmenttime]','$\_POST[app\_reason]','Pending','$\_POST[department]','$\_POST[doct]')";

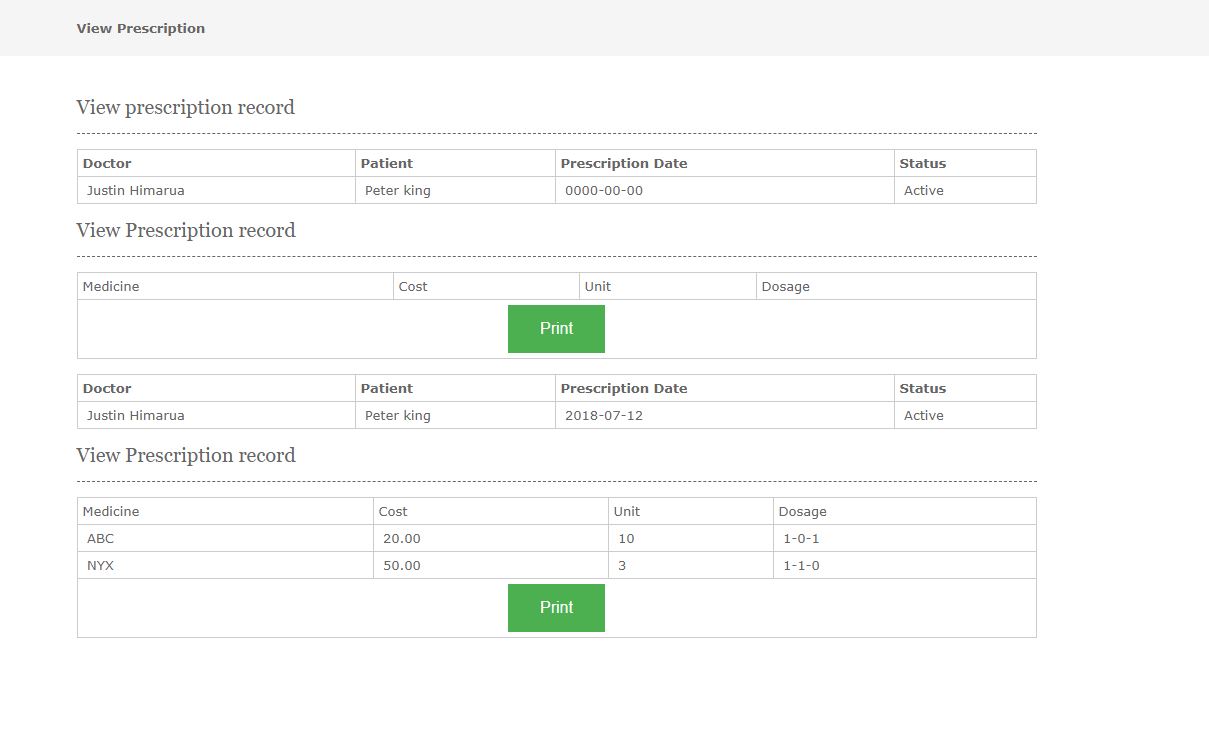
        if($qsql = mysqli\_query($con,$sql))

        {

            echo "<script>alert('Appointment record inserted successfully...');</script>";

        }

Webpage Queries (Print)



<script>

function myFunction()

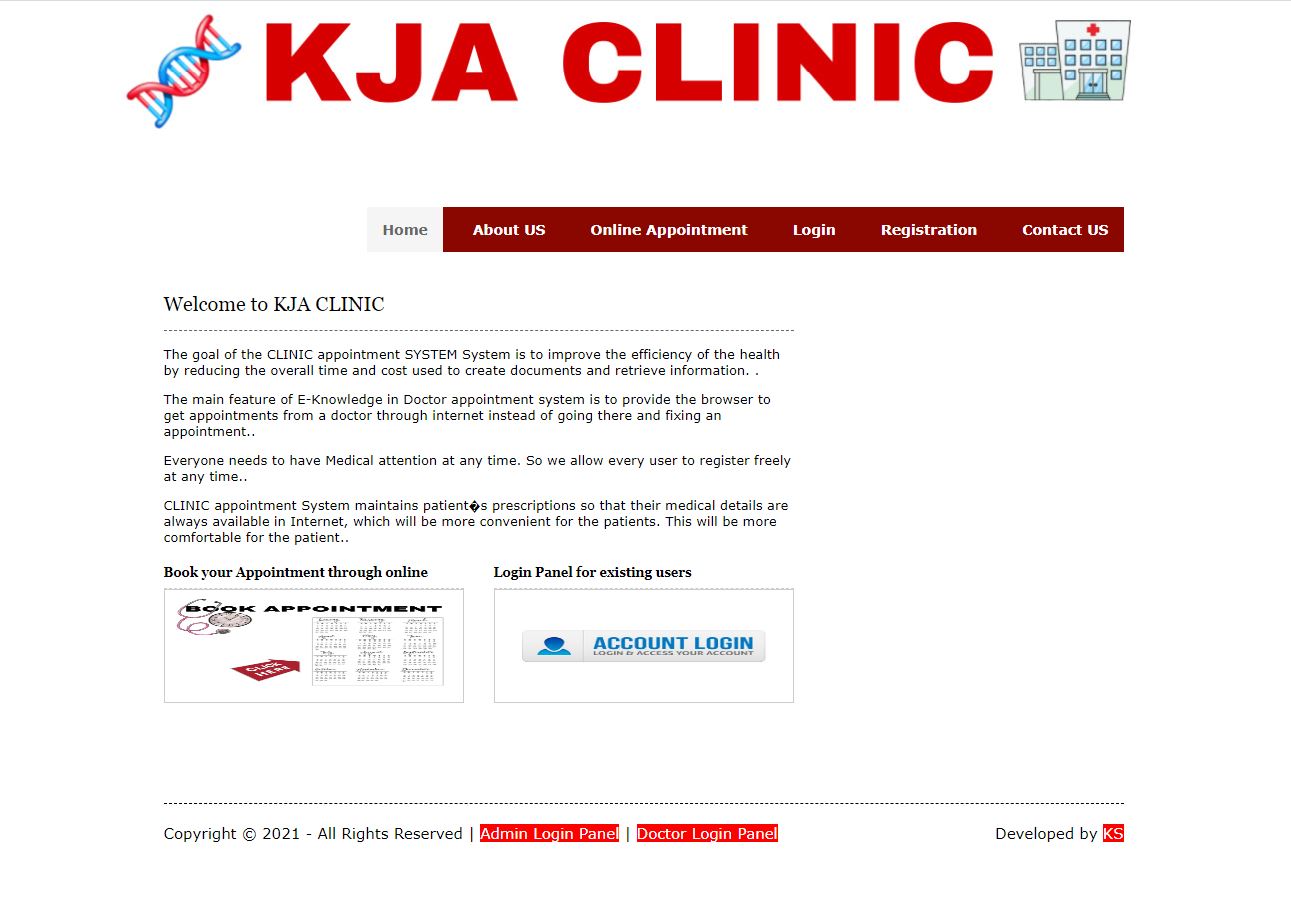
{

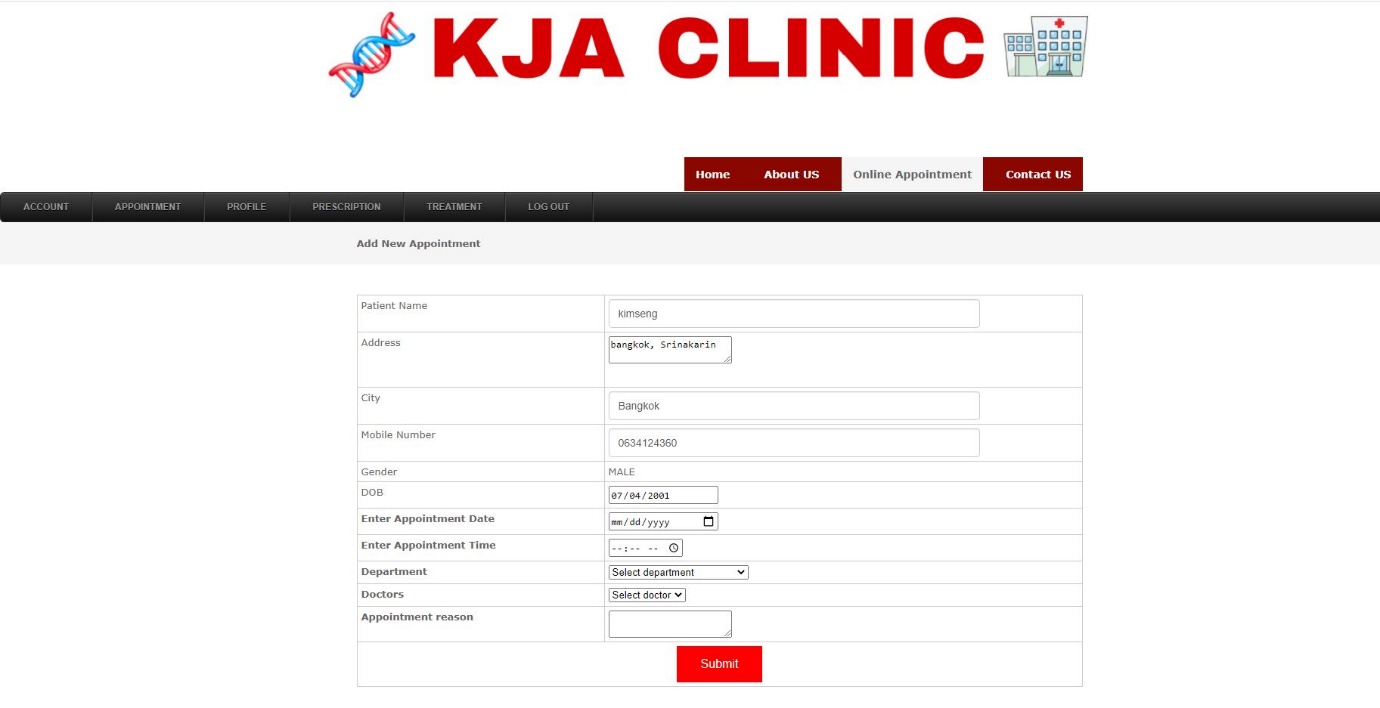
  window.print();

}

</script>

Our Demon





Future Work

* Improve as required to be more perfect
* Implement new web interface
* Add pay method that can pay online