TABLE OF CONTENTS

REQUIREMENT SPECIFICATIONS

Introduction

Hospital Management System is majorly to organise Patients, Employees and Services provided. Patients are the customers of the Hospital Management System, every data corresponding to the Patients must be stored and they must be provided with Services or Medication requested.

Working

The flow of Hospital Management System is as below
There are three categories of persons who enter the Hospital.
Patient, the person who is entering the Hospital for medical treatment.
Visitor, the person who is entering the Hospital for visiting other patients.
Emergency Patient, the person who is entering the Hospital with dire need of medical attention. In the case of an Emergency Patient the information collected need not be collected with the information, they are assigned with an id and rushed to the emergency ward. The patient can either avail the services which require no doctor prescription, provided by the hospital like Blood test, Urine test, Coronavirus test, ECG, X-Ray, Medicines etc or book an appointment to visit the doctor. The appointment booking will be for a doctor who is an expert in the area in which the patient requires consultation. The doctor then consults the patient either asking him to avail the services like MRI Scanning, CT Scanning, X-Ray, Blood test, medicines or in more serious situations perform surgery on the patient. The details regarding the surgeries are also stored. The surgeries are scheduled in operation theatres, so the assignment of an operation theatre to a patient is very crucial. After the surgery the patient is admitted into admit rooms and advised to stay in the admit rooms for a certain period of time before discharging from the hospital. Emergency patients and surgery undergoing patients are attended by the nurses. Who are the employees of the hospital like doctors? Employees are the backbone of the hospital.

The commercial part of the Hospital Management System relies on three major commodities. Services availment, Surgery payments. After the

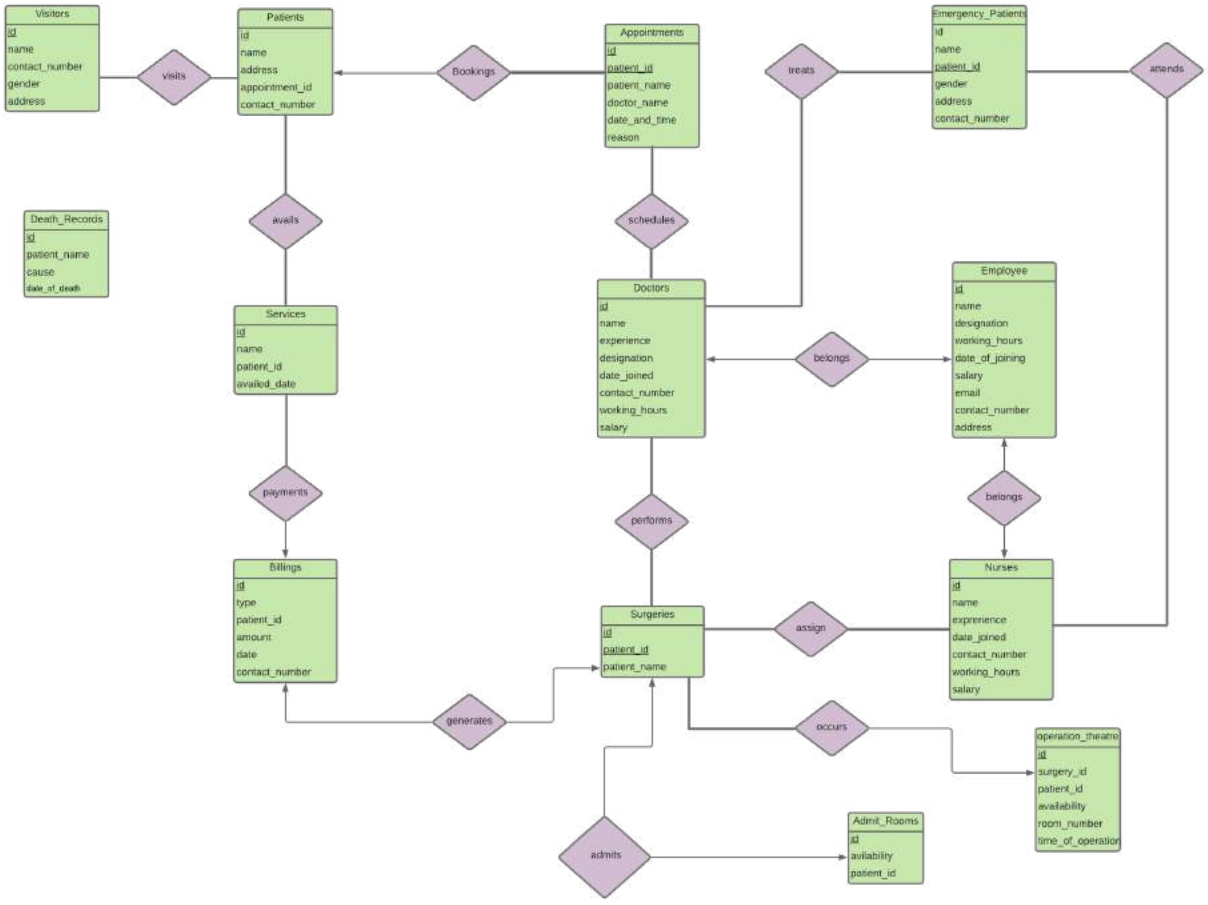
patient avails the services provided by the hospital, he needs to make the payment. Also, bills generated from the surgery of the patient need to be paid. It is also the hospital's responsibility to store the records of patients who are deceased.

Requirement

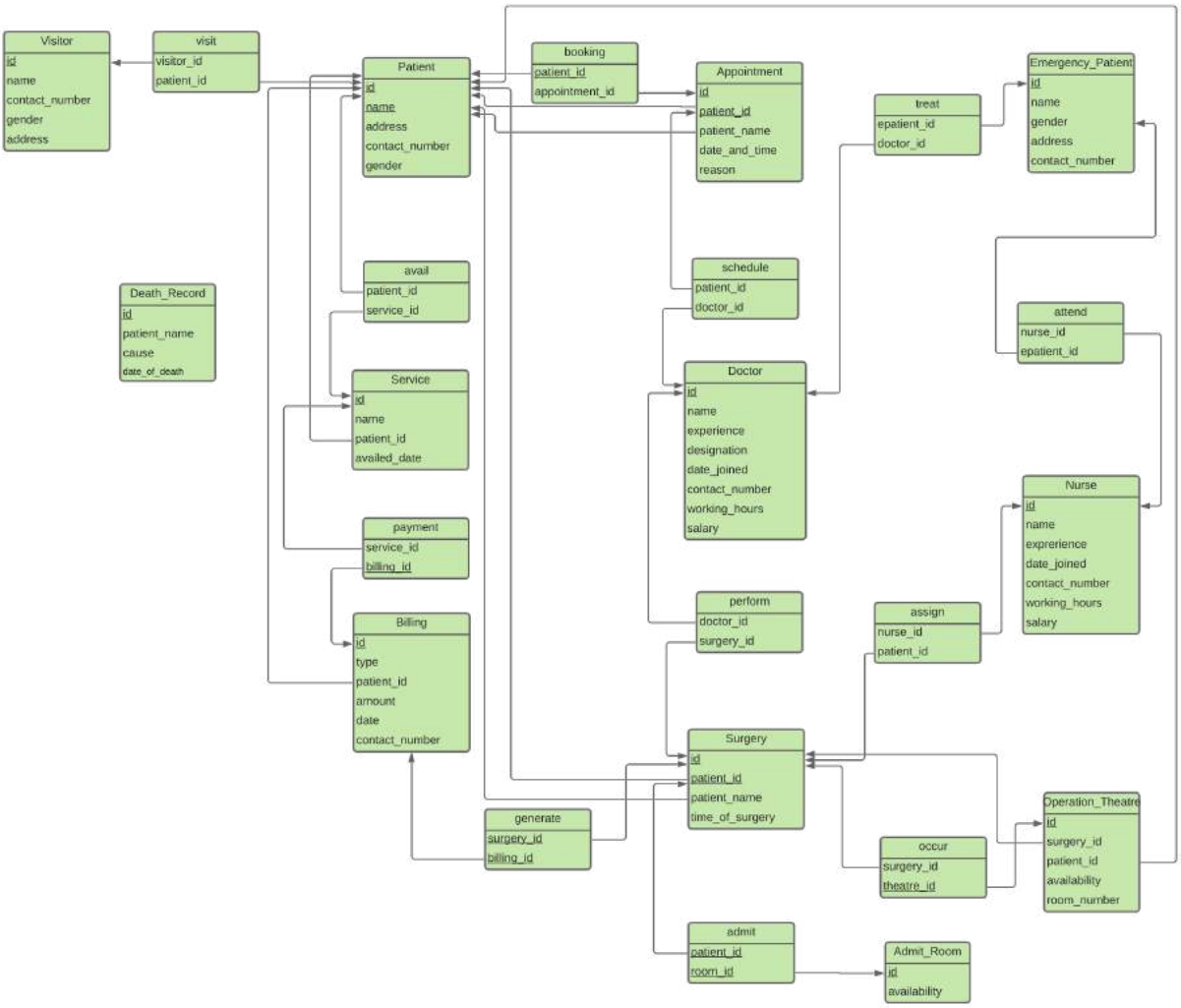
There is a requirement to store the data corresponding to
Patients, the persons who enter the hospital intending to get medical attention or avail any non-prescribed services. So, that the further procedures of the patient can be kept track of starting from here. Their details are to be stored.
Visitors, the persons who enter the hospital intending to visit patients in the hospital. Each visitor is given a visitor id, so that they are restricted to visit only the patient they are related to and only at certain periods of time. Their details such as contact number, address are collected for storing in case of any mishappening related to the visitor and the patient they visit.
Storing the details of emergency patients is another key requirement. But the patients need not provide the details prior to their admit into the hospital, they or their guardians can provide the hospital with the details post treatment.
Services like Blood test, Urine test, Coronavirus test, ECG, MRI scanning, CT scanning, X-Ray etc availed by each patient is to be stored. So, that each patient can be linked to the tests and find out the results corresponding to the test. The doctors can also prescribe any test which the patient then avails, and provides the doctor with reports. The doctor can make the diagnosis or perform any surgery based on the reports.
The surgery details are stored with the corresponding patient id, there may be a lot of doctors operating on a single patient. The complete data of the doctors operated on the patient in the surgery is to be stored.
Billing details are the commercial aspect of the hospital. Bill corresponding to each service, surgery is to be stored along with the patient id. This way total payment of each patient can be kept track of.
Employees data is also to be stored, to provide them their salary on time and keep track of their information for the times of any mishappening.

Rooms are intended to be of two types, operation theatres and admit rooms, the patient undergoing a surgery is related to the operation theatre in which their surgery takes place and the admit room where he is admitted after the surgery or in case of any mild medication. The details corresponding to the rooms are to be stored. Also, the details of patients who died due to unfortunate circumstances in the hospital are recorded and stored.

ENTITY RELATIONSHIP DIAGRAM



SCHEMA DIAGRAM



SCHEMA

There are total 24 tables in the schema

```
MariaDB [hospital]> show tables;
+--------------------+
| Tables_in_hospital |
+--------------------+
| Admit_Room         |
| Appointment        |
| Billing            |
| Death_Record       |
| Doctor             |
| Emergency_Patient  |
| Nurse              |
| Operation_Theatre  |
| Patient            |
| Service            |
| Surgery            |
| Visitor            |
| admit              |
| assign             |
| attend             |
| avail              |
| booking            |
| generate           |
| occur              |
| payment            |
| perform            |
| schedule           |
| treat              |
| visit              |
+--------------------+
24 rows in set (0.003 sec)
```

Table Name: Admit_Room

```
MariaDB [hospital]> describe Admit_Room;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| id           | varchar(10) | NO   | PRI | NULL    |       |
| availability | tinyint(1)  | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
2 rows in set (0.007 sec)
```

Table Name: Appointment

```
MariaDB [hospital]> describe Appointment;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| id           | varchar(10) | NO   | PRI | NULL    |       |
| patient_id   | varchar(10) | NO   | PRI | NULL    |       |
| patient_name | varchar(50) | YES  |     | NULL    |       |
| date_and_time | datetime   | YES  |     | NULL    |       |
| reason       | varchar(80) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.007 sec)
```

Table Name: Billing

```
MariaDB [hospital]> describe Billing;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| id            | varchar(10) | NO   | PRI | NULL    |       |
| type          | varchar(10) | YES  |     | NULL    |       |
| patient_id    | varchar(10) | YES  | MUL | NULL    |       |
| amount        | float       | YES  |     | NULL    |       |
| date          | datetime    | YES  |     | NULL    |       |
| contact_number | varchar(10) | YES |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
6 rows in set (0.010 sec)
```

Table Name: Death_Record

```
MariaDB [hospital]> describe Death_Record;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| id            | varchar(10) | NO   | PRI | NULL    |       |
| patient_name  | varchar(50) | YES  |     | NULL    |       |
| cause         | varchar(50) | YES  |     | NULL    |       |
| date_of_death | datetime    | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
4 rows in set (0.009 sec)
```

Table Name: Doctor

```
MariaDB [hospital]> describe Doctor;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| id             | varchar(10) | NO   | PRI | NULL    |       |
| name           | varchar(50) | YES  |     | NULL    |       |
| experience     | float       | YES  |     | NULL    |       |
| designation    | varchar(20) | YES  |     | NULL    |       |
| date_joined    | date        | YES  |     | NULL    |       |
| contact_number | varchar(10) | YES  |     | NULL    |       |
| working_hours  | float       | YES  |     | NULL    |       |
| salary         | float       | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
8 rows in set (0.013 sec)
```

Table Name: Emergency_Patient

```
MariaDB [hospital]> describe Emergency_Patient;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| id             | varchar(10) | NO   | PRI | NULL    |       |
| name           | varchar(50) | YES  |     | NULL    |       |
| gender         | varchar(10) | YES  |     | NULL    |       |
| address        | varchar(80) | YES  |     | NULL    |       |
| contact_number | varchar(10) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
5 rows in set (0.015 sec)
```

Table Name: Nurse

```
MariaDB [hospital]> describe Nurse;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| id             | varchar(10) | NO   | PRI | NULL    |       |
| name           | varchar(50) | YES  |     | NULL    |       |
| experience     | float       | YES  |     | NULL    |       |
| date_joined    | date        | YES  |     | NULL    |       |
| contact_number | varchar(10) | YES  |     | NULL    |       |
| working_hours  | float       | YES  |     | NULL    |       |
| salary         | float       | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
7 rows in set (0.011 sec)
```

Table Name: Operation_Theatre

```
MariaDB [hospital]> describe Operation_Theatre;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| id           | varchar(10) | NO   | PRI | NULL    |       |
| surgery_id   | varchar(10) | YES  | MUL | NULL    |       |
| patient_id   | varchar(10) | YES  | MUL | NULL    |       |
| availability | tinyint(1)  | YES  |     | NULL    |       |
| room_number  | int(10)     | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.008 sec)
```

Table Name: Patient

```
MariaDB [hospital]> describe Patient;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| id             | varchar(10) | NO   | PRI | NULL    |       |
| name           | varchar(50) | NO   | PRI | NULL    |       |
| address        | varchar(80) | YES  |     | NULL    |       |
| contact_number | varchar(10) | YES  |     | NULL    |       |
| gender         | varchar(10) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
5 rows in set (0.010 sec)
```

Table Name: Service

```
MariaDB [hospital]> describe Service;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| id          | varchar(10) | NO   | PRI | NULL    |       |
| name        | varchar(50) | YES  |     | NULL    |       |
| patient_id  | varchar(10) | YES  | MUL | NULL    |       |
| availed_date | datetime    | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.006 sec)
```

Table Name: Surgery

```
MariaDB [hospital]> describe Surgery;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| id              | varchar(10) | NO   | PRI | NULL    |       |
| patient_id      | varchar(10) | NO   | PRI | NULL    |       |
| patient_name    | varchar(50) | YES  |     | NULL    |       |
| time_of_surgery | datetime    | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
4 rows in set (0.008 sec)
```

Table Name: Visitor

```
MariaDB [hospital]> describe Visitor;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| id             | varchar(10) | NO   | PRI | NULL    |       |
| name           | varchar(50) | YES  |     | NULL    |       |
| contact_number | varchar(10) | YES  |     | NULL    |       |
| gender         | varchar(10) | YES  |     | NULL    |       |
| address        | varchar(80) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
5 rows in set (0.010 sec)
```

Table Name: admit

```
MariaDB [hospital]> describe admit;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| patient_id | varchar(10) | NO   | PRI | NULL    |       |
| room_id    | varchar(10) | NO   | PRI | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.011 sec)
```

Table Name: assign

```
MariaDB [hospital]> describe assign;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| nurse_id   | varchar(10) | YES  | MUL | NULL    |       |
| patient_id | varchar(10) | YES  | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.008 sec)
```

Table Name: attend

```
MariaDB [hospital]> describe attend;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| nurse_id    | varchar(10) | YES  | MUL | NULL    |       |
| epatient_id | varchar(10) | YES  | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
2 rows in set (0.007 sec)
```

Table Name: avail

```
MariaDB [hospital]> describe avail;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| patient_id | varchar(10) | YES  | MUL | NULL    |       |
| service_id | varchar(10) | YES  | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.008 sec)
```

Table Name: booking

```
MariaDB [hospital]> describe booking;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| patient_id     | varchar(10) | NO   | PRI | NULL    |       |
| appointment_id | varchar(10) | YES  | MUL | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
2 rows in set (0.008 sec)
```

Table Name: generate

```
MariaDB [hospital]> describe generate;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| surgery_id | varchar(10) | NO   | PRI | NULL    |       |
| billing_id | varchar(10) | NO   | PRI | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.007 sec)
```

Table Name: occur

```
MariaDB [hospital]> describe occur;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| surgery_id | varchar(10) | YES  | MUL | NULL    |       |
| theatre_id | varchar(10) | NO   | PRI | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.008 sec)
```

Table Name: payment

```
MariaDB [hospital]> describe payment;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| service_id | varchar(10) | YES  | MUL | NULL    |       |
| billing_id | varchar(10) | NO   | PRI | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.007 sec)
```

Table Name: perform

```
MariaDB [hospital]> describe perform;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| doctor_id  | varchar(10) | YES  | MUL | NULL    |       |
| surgery_id | varchar(10) | YES  | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.007 sec)
```

Table Name: schedule

```
MariaDB [hospital]> describe schedule;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| patient_id | varchar(10) | YES  | MUL | NULL    |       |
| doctor_id  | varchar(10) | YES  | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.007 sec)
```

Table Name: treat

```
MariaDB [hospital]> describe treat;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| epatient_id | varchar(10) | YES  | MUL | NULL    |       |
| doctor_id   | varchar(10) | YES  | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
2 rows in set (0.013 sec)
```

Table Name: visit

```
MariaDB [hospital]> describe visit;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| visitor_id | varchar(10) | YES  | MUL | NULL    |       |
| patient_id | varchar(10) | YES  | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.009 sec)
```

KEY AREAS OF SCHEMA

1. Patients who have taken Doctor appointment

```
MariaDB [hospital]> SELECT id, name
    -> FROM Patient
    -> WHERE id IN (SELECT patient_id from Appointment);
+-----------+-----------------------+
| id        | name                  |
+-----------+-----------------------+
| 111801001 | Captain Jack Sparrow  |
| 111801002 | Gellert Grindelwald   |
| 111801003 | Elizabeth Swann       |
| 111801004 | Harry Potter          |
| 111801005 | Ronald Weasley        |
| 111801006 | Hermione Granger      |
| 111801007 | Albus Dumbledore      |
| 111801029 | Tom Riddle            |
| 111801031 | Sirius Black          |
| 111801034 | Neville Longbottom    |
| 111801045 | Draco Malfoy          |
+-----------+-----------------------+
11 rows in set (0.007 sec)
```

2. Visitors who came to visit Patients

```
MariaDB [hospital]> SELECT Visitor.name, Patient.name
    -> FROM Visitor, Patient, visit
    -> WHERE visit.visitor_id = Visitor.id AND visit.patient_id = Patient.id;
+-------------+-----------------------+
| name        | name                  |
+-------------+-----------------------+
| Will Turner | Captain Jack Sparrow  |
| Will Turner | Gellert Grindelwald   |
| James Norry | Captain Jack Sparrow  |
| Hector      | Elizabeth Swann       |
| Hector      | Harry Potter          |
| Ragetti     | Elizabeth Swann       |
| Ana Maria   | Ronald Weasley        |
| Ana Maria   | Hermione Granger      |
| Joshamee    | Ronald Weasley        |
| Marty       | Tom Riddle            |
| Mull roy    | Sirius Black          |
| Murtogg     | Neville Longbottom    |
| Lieutenant  | Draco Malfoy          |
+-------------+-----------------------+
13 rows in set (0.001 sec)
```

3. Patients who received surgery along with the doctor who performed surgery and time of surgery arranged in chronological order

```
MariaDB [hospital]> SELECT patient_id, patient_name, Doctor.name AS Doctor_name, time_of_surgery
    -> FROM Surgery, perform, Doctor
    -> WHERE Surgery.id = perform.surgery_id AND perform.doctor_id = Doctor.id
    -> ORDER BY time_of_surgery ASC;
+------------+--------------------+---------------------+---------------------+
| patient_id | patient_name       | Doctor_name         | time_of_surgery     |
+------------+--------------------+---------------------+---------------------+
| 111801031  | Sirius Black       | Charles Richard Drew | 1980-04-15 13:44:00 |
| 111801045  | Draco Malfoy       | Charles Richard Drew | 1990-05-28 19:36:28 |
| 111801034  | Neville Longbottom | Georges Mathe       | 1999-05-14 10:17:02 |
| 111801029  | Tom Riddle         | Elizabeth Blackwell | 2004-09-23 23:37:27 |
| 111801004  | Harry Potter       | Helene D.Gayle      | 2017-05-24 14:35:42 |
| 111801002  | Gellert Grindelwald | Helene D.Gayle      | 2017-11-02 21:18:31 |
| 111801005  | Ronald Weasley     | Edward Jenner       | 2017-11-20 00:14:34 |
+------------+--------------------+---------------------+---------------------+
7 rows in set (0.001 sec)
```

4. Patients who availed any service, the type of the service availed and the amount of bill generated corresponding to the service ordered descendingly according to the amount(highest bill comes on top).

```
MariaDB [hospital]> SELECT Patient.name, Service.name, Billing.amount
    -> FROM Patient, Service, Billing, avail, payment
    -> WHERE Patient.id = avail.patient_id AND avail.service_id = Service.id
    ->                                    AND Service.id = payment.service_id
    ->                                    AND payment.billing_id = Billing.id;
+----------------------+-----------------+----------+
| name                 | name            | amount   |
+----------------------+-----------------+----------+
| Captain Jack Sparrow | Blood test      | 2480.02  |
| Elizabeth Swann      | X-Ray test      |    5480  |
| Gellert Grindelwald  | HRCT test       |    2000  |
| Albus Dumbledore     | Medical Checkup | 17845.5  |
| Ronald Weasley       | ENT             |   15480  |
| Tom Riddle           | Blood test      | 2480.02  |
| Sirius Black         | Medical Checkup | 17845.5  |
+----------------------+-----------------+----------+
7 rows in set (0.002 sec)
```

5. Nurses assigned to surgery undergone patients

```
MariaDB [hospital]> SELECT Nurse.name, Surgery.patient_name
    -> FROM Nurse, Surgery, assign
    -> WHERE Surgery.patient_id = assign.patient_id AND assign.nurse_id = Nurse.id;
+-----------------+---------------------+
| name            | patient_name        |
+-----------------+---------------------+
| Julie Watson    | Harry Potter        |
| Emily Parker    | Ronald Weasley      |
| Jamie Rose      | Tom Riddle          |
| Tiffany Morrison | Sirius Black       |
| Sophie Jane     | Neville Longbottom  |
+-----------------+---------------------+
5 rows in set (0.002 sec)
```

6. Name, bill amount and bill type generated by surgery undergone by patients.

```
MariaDB [hospital]> SELECT patient_name, amount, type
    -> FROM Surgery, Billing, generate
    -> WHERE Surgery.id = generate.surgery_id AND generate.billing_id = Billing.id;
+---------------------+----------+-------------+
| patient_name        | amount   | type        |
+---------------------+----------+-------------+
| Gellert Grindelwald | 17845.5  | UPI         |
| Harry Potter        | 2480.02  | Cash        |
| Ronald Weasley      |    5480  | UPI         |
| Tom Riddle          |    2000  | Debit Card  |
| Sirius Black        | 17845.5  | Cash        |
| Neville Longbottom  |   15480  | Debit Card  |
| Draco Malfoy        | 2480.02  | Cash        |
+---------------------+----------+-------------+
7 rows in set (0.003 sec)
```

7. Name of the patient who undergone surgery, the room in which operation took place, the time of operation and the room in which the patient was admitted after the surgery

```
MariaDB [hospital]> SELECT Patient.name, room_number AS operation_room_number,
    ->                    time_of_surgery, Admit_Room.id AS admit_room_number
    -> FROM Operation_Theatre, Surgery, admit, Admit_Room, Patient
    -> WHERE Operation_Theatre.patient_id = Patient.id AND Operation_Theatre.surgery_id = Surgery.id
    ->  AND  Surgery.patient_id = admit.patient_id AND admit.room_id = Admit_Room.id;
+--------------------+-----------------------+---------------------+--------------------+
| name               | operation_room_number | time_of_surgery     | admit_room_number |
+--------------------+-----------------------+---------------------+--------------------+
| Harry Potter       |                   108 | 2017-05-24 14:35:42 | 101                |
| Ronald Weasley     |                   109 | 2017-11-20 00:14:34 | 102                |
| Tom Riddle         |                   208 | 2004-09-23 23:37:27 | 103                |
| Sirius Black       |                   209 | 1980-04-15 13:44:00 | 104                |
| Draco Malfoy       |                   210 | 1990-05-28 19:36:28 | 105                |
| Gellert Grindelwald |                  208 | 2017-11-02 21:18:31 | 106                |
+--------------------+-----------------------+---------------------+--------------------+
6 rows in set (0.001 sec)
```

8. Death records is an independent table which has no foreign key references to any other table in the database

```
MariaDB [hospital]> SELECT * FROM Death_Record;
+------+----------------+----------------+---------------------+
| id   | patient_name   | cause          | date_of_death       |
+------+----------------+----------------+---------------------+
| 1032 | Adalberto Dray | Heart Attack   | 1995-05-21 00:00:00 |
| 1045 | Hilda Flanery  | Accident       | 1990-04-12 00:00:00 |
| 1154 | Eve Rampton    | Kidneys Failure | 1985-12-14 00:00:00 |
| 1561 | June Terhune   | Accident       | 1997-09-05 00:00:00 |
| 1847 | Nguyet Dutra   | Blood Cancer   | 1983-11-25 00:00:00 |
+------+----------------+----------------+---------------------+
5 rows in set (0.000 sec)
```

9. Doctors who treated the emergency patients and nurses attended them

```
MariaDB [hospital]> SELECT Emergency_Patient.name AS patient_name, Doctor.name as doctor_name,
    ->                     Nurse.name AS nurse_name
    -> FROM Emergency_Patient, Doctor, treat, Nurse, attend
    -> WHERE Emergency_Patient.id = treat.epatient_id AND treat.doctor_id = Doctor.id
    ->  AND  Emergency_Patient.id = attend.epatient_id AND attend.nurse_id = Nurse.id;
+-----------------+----------------------+-----------------+
| patient_name    | doctor_name          | nurse_name      |
+-----------------+----------------------+-----------------+
| Johnson Bravo   | Edward Jenner        | Tiffany Morrison |
| Olivia          | Helene D.Gayle       | Julie Watson    |
| George Smith    | Virginia Apgar       | Stephany Johnson |
| Olivia Morris   | Edward Jenner        | Emily Parker    |
| Jenson Nicolson | Charles Richard Drew | Tiffany Morrison |
| Jimmy williams  | Virginia Apgar       | Julie Watson    |
| Mitchell Santner | Edward Jenner       | Marie Phillips  |
+-----------------+----------------------+-----------------+
7 rows in set (0.008 sec)
```

10. Revenue generated by the hospital so far, rounded to the nearest integer.

```
MariaDB [hospital]> SELECT ROUND(SUM(amount), 0) AS revenue
    -> FROM Billing;
+---------+
| revenue |
+---------+
|  127222 |
+---------+
1 row in set (0.001 sec)
```

VIEWS

1. View Name: patient_appointment_service

```
MariaDB [hospital]> DESCRIBE patient_appointment_service;
+---------------------------+-------------+------+-----+---------+-------+
| Field                     | Type        | Null | Key | Default | Extra |
+---------------------------+-------------+------+-----+---------+-------+
| patient_id                | varchar(10) | NO   |     | NULL    |       |
| patient_name              | varchar(50) | NO   |     | NULL    |       |
| appointment_reason        | varchar(80) | YES  |     | NULL    |       |
| appointment_date_and_time | datetime    | YES  |     | NULL    |       |
| service_name              | varchar(50) | YES  |     | NULL    |       |
| service_availed_date      | datetime    | YES  |     | NULL    |       |
+---------------------------+-------------+------+-----+---------+-------+
6 rows in set (0.012 sec)
```

Justification:
If a patient enters hospital, then his motive can be to book an appointment or avail a service. This view captures the motive of all the patients entering the hospital. Statistics can be calculated according to the view about the number of people availing service or appointment or both.
Later these statistics can be used for the further development of the hospital system. Also, doctors can look at the past medical tests availed by a patient as a service, and recommend any future tests.

2. View Name: Billing_Service_Surgery

```
MariaDB [hospital]> DESCRIBE Billing_Service_Surgery;
+---------------------+-------------+------+-----+---------+-------+
| Field               | Type        | Null | Key | Default | Extra |
+---------------------+-------------+------+-----+---------+-------+
| billing_id          | varchar(10) | NO   |     | NULL    |       |
| patient_id          | varchar(10) | YES  |     | NULL    |       |
| billing_amount      | float       | YES  |     | NULL    |       |
| billing_date        | datetime    | YES  |     | NULL    |       |
| service_id          | varchar(10) | YES  |     | NULL    |       |
| service_name        | varchar(50) | YES  |     | NULL    |       |
| service_availed_date | datetime    | YES  |     | NULL    |       |
| surgery_id          | varchar(10) | YES  |     | NULL    |       |
| time_of_surgery     | datetime    | YES  |     | NULL    |       |
+---------------------+-------------+------+-----+---------+-------+
9 rows in set (0.006 sec)
```

Justification:
Revenue is very important for a hospital management system. Also, the statistics involved in the revenue generation.
This view helps to know from where the majority of the revenue is generated, from services offered or from surgeries performed.
At the same time an entire statistics of the revenue from service and surgery, provides insights to the hospital about the expenditure.
Every billing is made under patient id, so a patient can know the amount of money spent by him, split into categories.

3. View Name: patient_epatient

```
MariaDB [hospital]> DESCRIBE patient_epatient;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| patient_id   | varchar(10) | NO   |     |         |       |
| patient_name | varchar(50) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
2 rows in set (0.006 sec)
```

Justification:
There are two types of patients in the hospital, normal patients and emergency patients. The management needs to know the entire list of patients who are getting served, or treated in the hospital.
This view will help to provide statistics of the total people who are treated in the hospital. Take any action, if the count of patients is decreasing and increase capacity of the hospital, if the count of patients is increasing.

4. View Name: employee

```
MariaDB [hospital]> DESCRIBE employee;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| employee_id    | varchar(10) | NO   |     |         |       |
| employee_name  | varchar(50) | YES  |     | NULL    |       |
| working_hours  | float       | YES  |     | NULL    |       |
| employee_salary| float       | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
4 rows in set (0.007 sec)
```

Justification:
Management needs to keep track of all the employees, to provide them salaries. This view helps to give salary to employees, and keep track of the employees working in the hospital.

5. View Name: nurse_assign_surgery

```
MariaDB [hospital]> DESCRIBE nurse_assign_surgery;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| id           | varchar(10) | NO   |     | NULL    |       |
| patient_id   | varchar(10) | NO   |     | NULL    |       |
| patient_name | varchar(50) | YES  |     | NULL    |       |
| nurse_name   | varchar(50) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
4 rows in set (0.006 sec)
```

Justification:
To backtrack to nurses in case of any mishappening in the surgery period
Also to keep track of the working period of nurses

6. View Name: surgery_admit_admitRoom

```
MariaDB [hospital]> DESCRIBE surgery_admit_admitRoom;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| id           | varchar(10) | YES  |     | NULL    |       |
| patient_id   | varchar(10) | YES  |     | NULL    |       |
| patient_name | varchar(50) | YES  |     | NULL    |       |
| room_id      | varchar(10) | NO   |     | NULL    |       |
| availability | tinyint(1)  | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.005 sec)
```

Justification:
To know which patients are assigned to which rooms and to know the rooms which are empty so that it will helpful to assign new surgery patients

7. View Name: surgery_occur_operationTheatre

```
MariaDB [hospital]> DESCRIBE surgery_occur_operationTheatre;
+---------------------+------------+------+-----+---------+-------+
| Field               | Type       | Null | Key | Default | Extra |
+---------------------+------------+------+-----+---------+-------+
| id                  | varchar(10) | YES  |     | NULL    |       |
| patient_id          | varchar(10) | YES  |     | NULL    |       |
| patient_name        | varchar(50) | YES  |     | NULL    |       |
| operation_theatre_id | varchar(10) | NO   |     | NULL    |       |
| availability        | tinyint(1)  | YES  |     | NULL    |       |
+---------------------+------------+------+-----+---------+-------+
5 rows in set (0.007 sec)
```

Justification:
To know which patients are assigned to which operation theatres and to know the theatre rooms which are empty so that it will helpful to assign new surgery patients to do the operation

8. View Name: nurse_assign_patient

```
MariaDB [hospital]> DESCRIBE nurse_assign_patient;
+-------------+------------+------+-----+---------+-------+
| Field       | Type       | Null | Key | Default | Extra |
+-------------+------------+------+-----+---------+-------+
| nurse_id    | varchar(10) | NO   |     | NULL    |       |
| nurse_name  | varchar(50) | YES  |     | NULL    |       |
| patient_Id  | varchar(10) | NO   |     | NULL    |       |
| patient_name | varchar(50) | NO   |     | NULL    |       |
+-------------+------------+------+-----+---------+-------+
4 rows in set (0.005 sec)
```

Justification:
This will be a very frequently used query to find the nurses assigned to different persons.
If we want to know the nurses assigned to a particular patient we can perform queries in this view.

9. View Name: doctor_schedule_patient

```
MariaDB [hospital]> DESCRIBE doctor_schedule_patient;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| Doctor_id    | varchar(10) | NO   |     | NULL    |       |
| Doctor_name  | varchar(50) | YES  |     | NULL    |       |
| designation  | varchar(20) | YES  |     | NULL    |       |
| patient_id   | varchar(10) | NO   |     | NULL    |       |
| patient_name | varchar(50) | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.005 sec)
```

Justification:
This view shows the list of all the doctors treating the patients.
If we want to know the patients which are treated by a particular doctor.
We can perform queries in this view.

10. View Name: visitor_visit_patient

```
MariaDB [hospital]> DESCRIBE visitor_visit_patient;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| vistor_id    | varchar(10) | NO   |     | NULL    |       |
| visitor_name | varchar(50) | YES  |     | NULL    |       |
| vistor_number| varchar(10) | YES  |     | NULL    |       |
| patient_id   | varchar(10) | NO   |     | NULL    |       |
| patient_name | varchar(50) | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.011 sec)
```

Justification:
This view shows the list of visitors visiting the patients admitted in the hospital.
It will be easier to know the visitor who visited a particular patient by performing queries in this view.
There are various cases where we may need to look up into the list of visitors to a particular patient, in which case we can use pateint_id in this view to access the list of viewers directly in a simple query

11. View Name: doctor_epatient_nurse

```
MariaDB [hospital]> DESCRIBE doctor_epatient_nurse;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| doctor_name    | varchar(50) | YES  |     | NULL    |       |
| doctor_id      | varchar(10) | NO   |     | NULL    |       |
| e_patient_id   | varchar(10) | NO   |     | NULL    |       |
| e_patient_name | varchar(50) | YES  |     | NULL    |       |
| nurse_id       | varchar(10) | NO   |     | NULL    |       |
| nurse_name     | varchar(50) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
6 rows in set (0.008 sec)
```

Justification:
This is a useful view, just in case we wanna have a page landing at all the emergency patients list who have been admitted to the hospital along with the nurses and doctors who have taken up the case

12. View Name: doctor_surgery_nurse

```
MariaDB [hospital]> DESCRIBE doctor_surgery_nurse;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| doctor_id    | varchar(10) | NO   |     | NULL    |       |
| doctor_name  | varchar(50) | YES  |     | NULL    |       |
| surgery_id   | varchar(10) | NO   |     | NULL    |       |
| patient_id   | varchar(10) | NO   |     | NULL    |       |
| patient_name | varchar(50) | YES  |     | NULL    |       |
| nurse_id     | varchar(10) | NO   |     | NULL    |       |
| nurse_name   | varchar(50) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
7 rows in set (0.007 sec)
```

Justification:
Just like the previous case, we might need to look up on all the surgeries along with the doctors and nurses who were involved in the surgery as well. Thus in that case a doctor_surgery_nurse view can be helpful.

13. View Name: doctor_appointments

```
MariaDB [hospital]> DESCRIBE doctor_appointments;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| doctor_id      | varchar(10) | NO   |     | NULL    |       |
| doctor_name    | varchar(50) | YES  |     | NULL    |       |
| appointment_id | varchar(10) | NO   |     | NULL    |       |
| patient_name   | varchar(50) | YES  |     | NULL    |       |
| patient_id     | varchar(10) | NO   |     | NULL    |       |
| reason         | varchar(80) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
6 rows in set (0.006 sec)
```

Justification:
Suppose we want to display what are all the appointments scheduled to a particular doctor, life becomes easy with this view in such a case

14. View Name: patient_service

```
MariaDB [hospital]> DESCRIBE patient_service;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| patient_id   | varchar(10) | NO   |     | NULL    |       |
| patient_name | varchar(50) | NO   |     | NULL    |       |
| service_id   | varchar(10) | NO   |     | NULL    |       |
| service_name | varchar(50) | YES  |     | NULL    |       |
| date_availed | datetime    | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.007 sec)
```

Justification:
There are chances that we may need to look up all the services availed by some patient or data a particular service available by patient, in such a case, the patient_service view helps a lot to retrieve info in simple queries

FUNCTIONS

1. avg_salary

```
MariaDB [hospital]> DELIMITER #
MariaDB [hospital]> CREATE FUNCTION avg_salary()
    -> RETURNS FLOAT DETERMINISTIC
    -> BEGIN
    -> DECLARE value FLOAT;
    -> SET value = (SELECT AVG(salary) FROM Doctor);
    -> RETURN value;
    -> END; #
Query OK, 0 rows affected (0.052 sec)

MariaDB [hospital]> SELECT avg_salary()#
+--------------+
| avg_salary() |
+--------------+
|       109545 |
+--------------+
1 row in set (0.003 sec)
```

Justification: It keeps track of the average salary earning by the doctors in the hospital.

2. get_profit()

```
MariaDB [hospital]> DELIMITER #
MariaDB [hospital]> CREATE FUNCTION get_profit()
    -> RETURNS FLOAT DETERMINISTIC
    -> BEGIN
    -> DECLARE profit FLOAT;
    -> SET profit = (SELECT SUM(amount) FROM billing)
    ->            - (SELECT SUM(salary) FROM nurse)
    ->            - (SELECT SUM(salary) FROM doctor);
    -> RETURN profit;
    -> END; #
Query OK, 0 rows affected (0.071 sec)

MariaDB [hospital]> SELECT get_profit()#
+--------------+
| get_profit() |
+--------------+
|     -1193780 |
+--------------+
1 row in set (0.004 sec)
```

Justification: get_profit will be useful to see how much profit the hospital is in, it will show the status of the hospital financially. It will be able to judge how good or bad is the hospital doing in that particular location.

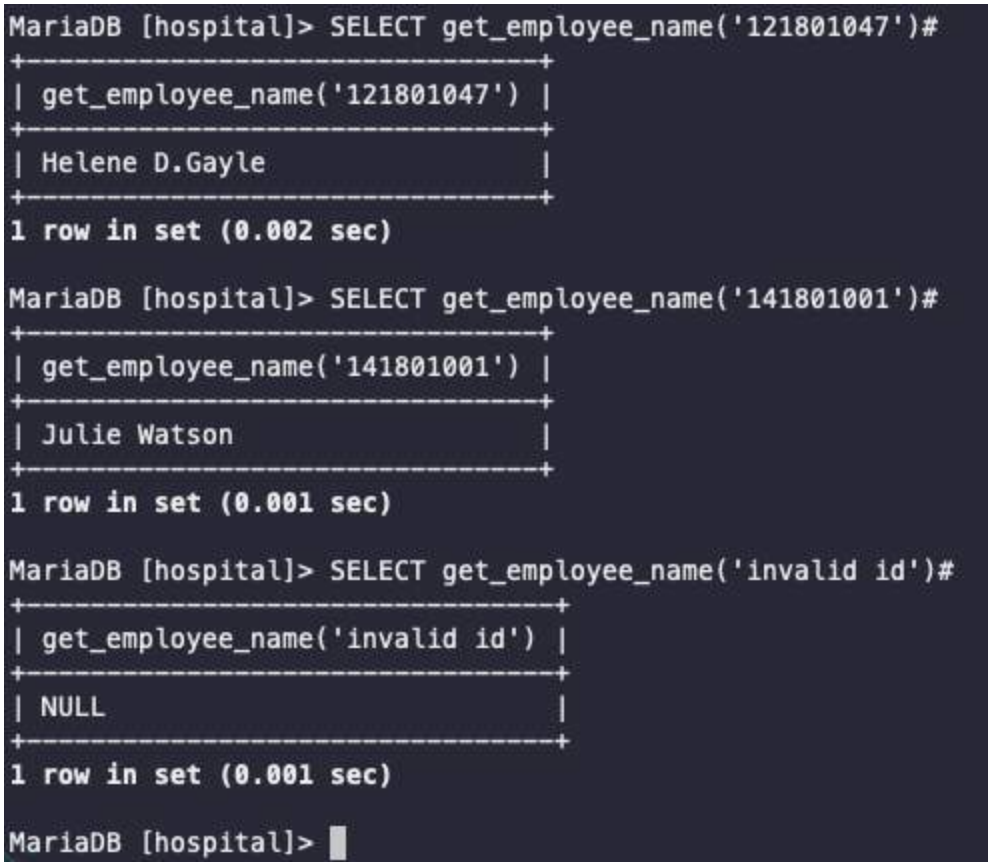
3. patient_admitted

```
MariaDB [hospital]> DELIMITER #
MariaDB [hospital]> CREATE FUNCTION patient_admitted(room_number INT)
    -> RETURNS VARCHAR(20) DETERMINISTIC
    -> BEGIN
    -> DECLARE name VARCHAR(20);
    -> SET name = (SELECT patient_name
    ->              FROM surgery
    ->              NATURAL JOIN
    ->              admit
    ->              WHERE room_id = room_number);
    -> RETURN name;
    -> END; #
Query OK, 0 rows affected (0.056 sec)

MariaDB [hospital]> SELECT patient_admitted(107)#
+------------------------+
| patient_admitted(107)  |
+------------------------+
| NULL                   |
+------------------------+
1 row in set (0.004 sec)
```

Justification: To know the patient name who is admitted in a room number given as input to the function patient_admitted.

4. get_employee_name

```
MariaDB [hospital]> DELIMITER #
MariaDB [hospital]> CREATE FUNCTION get_employee_name(eid VARCHAR(10))
    -> RETURNS VARCHAR(50) DETERMINISTIC
    -> BEGIN
    -> DECLARE ename VARCHAR(50);
    -> SELECT (CASE WHEN (eid IN (SELECT id FROM doctor)) THEN (SELECT name FROM doctor WHERE id=eid)
    ->              WHEN (eid IN (SELECT id FROM nurse)) THEN (SELECT name FROM nurse WHERE id=eid) END)
    -> INTO ename;
    -> RETURN ename;
    -> END;#
Query OK, 0 rows affected (0.055 sec)
```



Justification: if given an employee id, either belonging to doctor or nurse, we need to get their name. This might be useful in the case where we want to identify an employee just based on their id.

PROCEDURES

1. Doctor_Surgeries

```
MariaDB [hospital]>
MariaDB [hospital]> CREATE PROCEDURE Doctor_Surgeries()
    -> BEGIN
    -> SELECT Id FROM Doctor WHERE Id IN (SELECT doctor_id
    ->                                    FROm perform
    ->                                    GROUP BY (doctor_id)
    ->                                    HAVING COUNT(doctor_id)>=1);
    -> END ; #
Query OK, 0 rows affected (0.080 sec)

MariaDB [hospital]> call Doctor_Surgeries();
    -> #
+------------+
| Id         |
+------------+
| 121801045 |
| 121801046 |
| 121801047 |
| 121801048 |
| 121801051 |
+------------+
5 rows in set (0.007 sec)

Query OK, 0 rows affected (0.007 sec)
```

Justification: Doctor_surgeries returns the Id's of the doctors who have performed surgeries

2. get_waiting_list

```
MariaDB [hospital]> CREATE PROCEDURE get_waiting_list(IN d_id VARCHAR(15))
    -> BEGIN
    -> SELECT * FROM patient
    -> WHERE id IN (SELECT patient_id FROM schedule WHERE doctor_id = d_id);
    -> END;#
Query OK, 0 rows affected (0.081 sec)
```

Justification: get_waiting_list is always helpful for the doctors to see how many people are waiting for them, so that they can plan their next events. If there are no patients waiting in the queue, the doctor can go take a break. If there are many patients waiting for a doctor, he can decide his next events of the day and so on

3. patients_service_taken

```
MariaDB [hospital]> CREATE PROCEDURE patients_service_taken(IN service_name VARCHAR(15))
    -> BEGIN
    -> SELECT patient_ID FROM service WHERE name = service_name;
    -> END; #
Query OK, 0 rows affected (0.065 sec)
```

Justification: To get the list of all patients who had availed the service given as input to the procedure call patients_service_taken.

4. get_employee_name

```
MariaDB [hospital]> DELIMITER #
MariaDB [hospital]> CREATE PROCEDURE available_ot_and_ar()
    -> BEGIN
    -> SELECT o.room_number AS ot_room_number, a.id AS ar_room_number
    -> FROM Operation_Theatre AS o LEFT JOIN Admit_Room AS a
    -> ON o.availability = a.availability
    -> WHERE o.availability=0
    -> UNION
    -> SELECT o.room_number AS ot_room_number, a.id AS ar_room_number
    -> FROM Admit_room AS a RIGHT JOIN Operation_Theatre as o
    -> ON a.availability = o.availability
    -> WHERE a.availability=0;
    -> END; #
Query OK, 0 rows affected (0.079 sec)
```

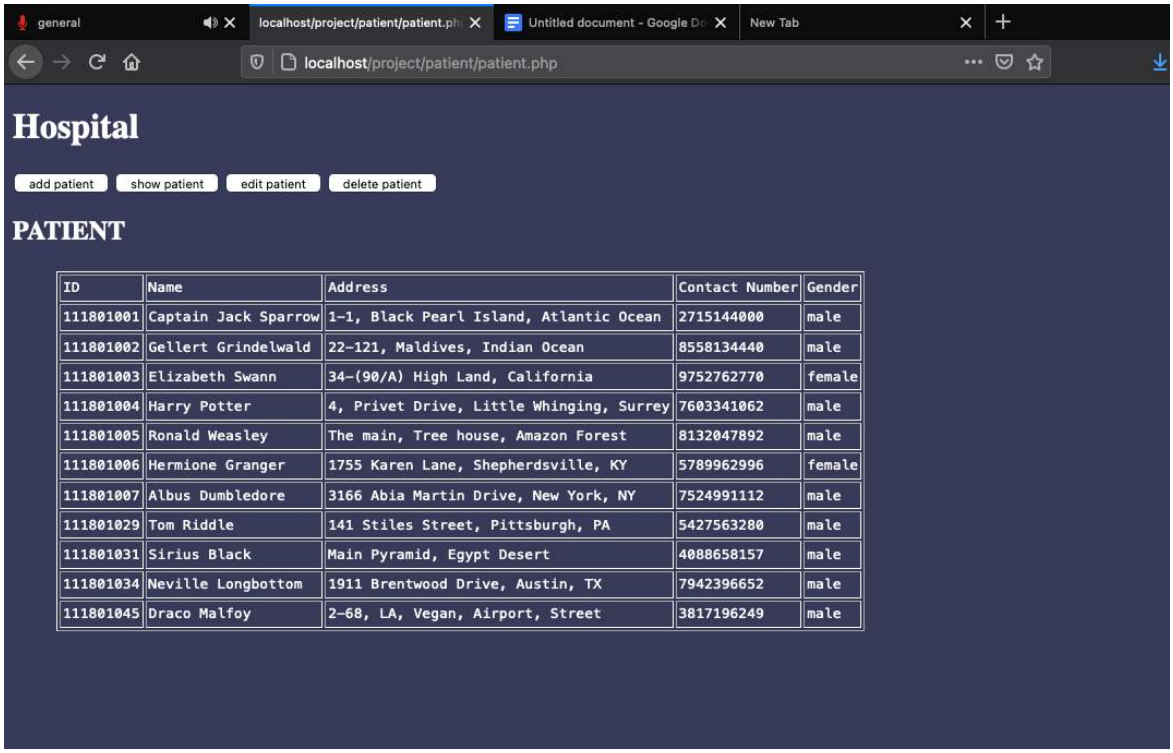
```
MariaDB [hospital]>
MariaDB [hospital]> call available_ot_and_ar;
+----------------+----------------+
| ot_room_number | ar_room_number |
+----------------+----------------+
|            210 | 107            |
|            210 | 201            |
|            210 | 202            |
|            210 | 203            |
|            210 | 204            |
|            210 | 205            |
|            210 | 206            |
|            210 | 207            |
+----------------+----------------+
8 rows in set (0.001 sec)

Query OK, 0 rows affected (0.002 sec)

MariaDB [hospital]>
```

Justification: when a surgery is to be conducted, the corresponding operation theatre room number and admit room room number is to be

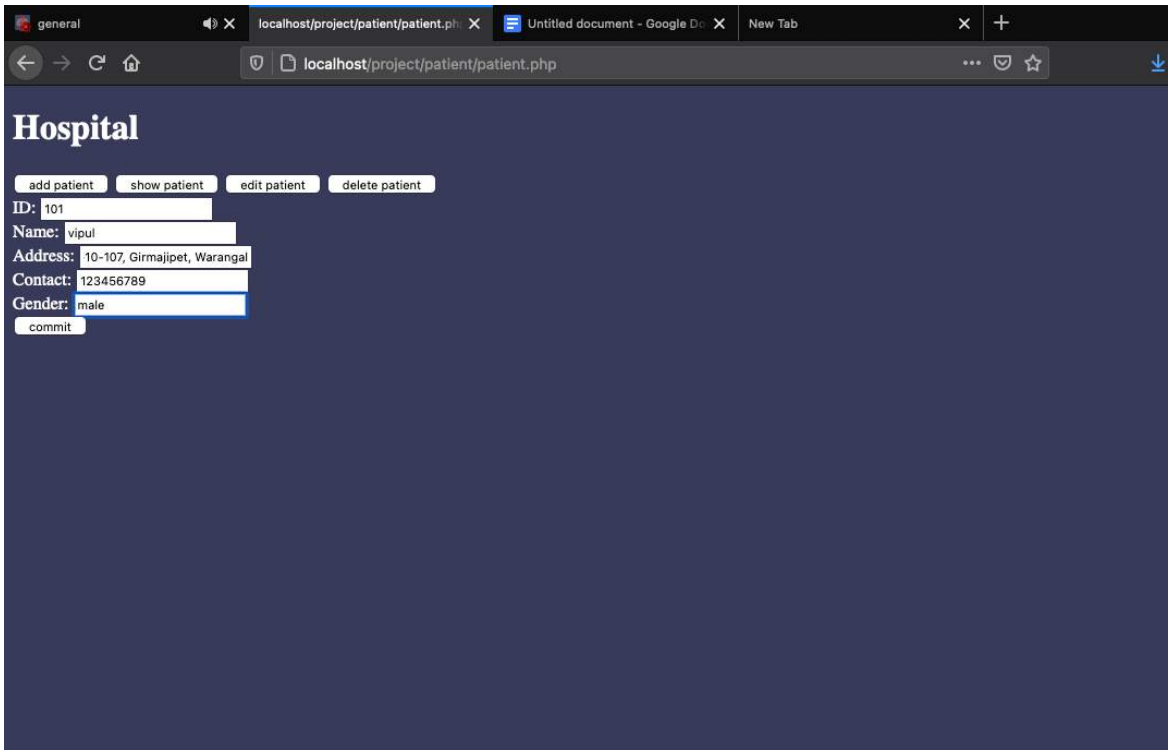
decided, to place the patient in those corresponding rooms. This procedure returns a combination of operation theatre(ot) room number and admit room(ar) room number which are currently available. Any combination can be frozen for a patient undergoing surgery beforehand.

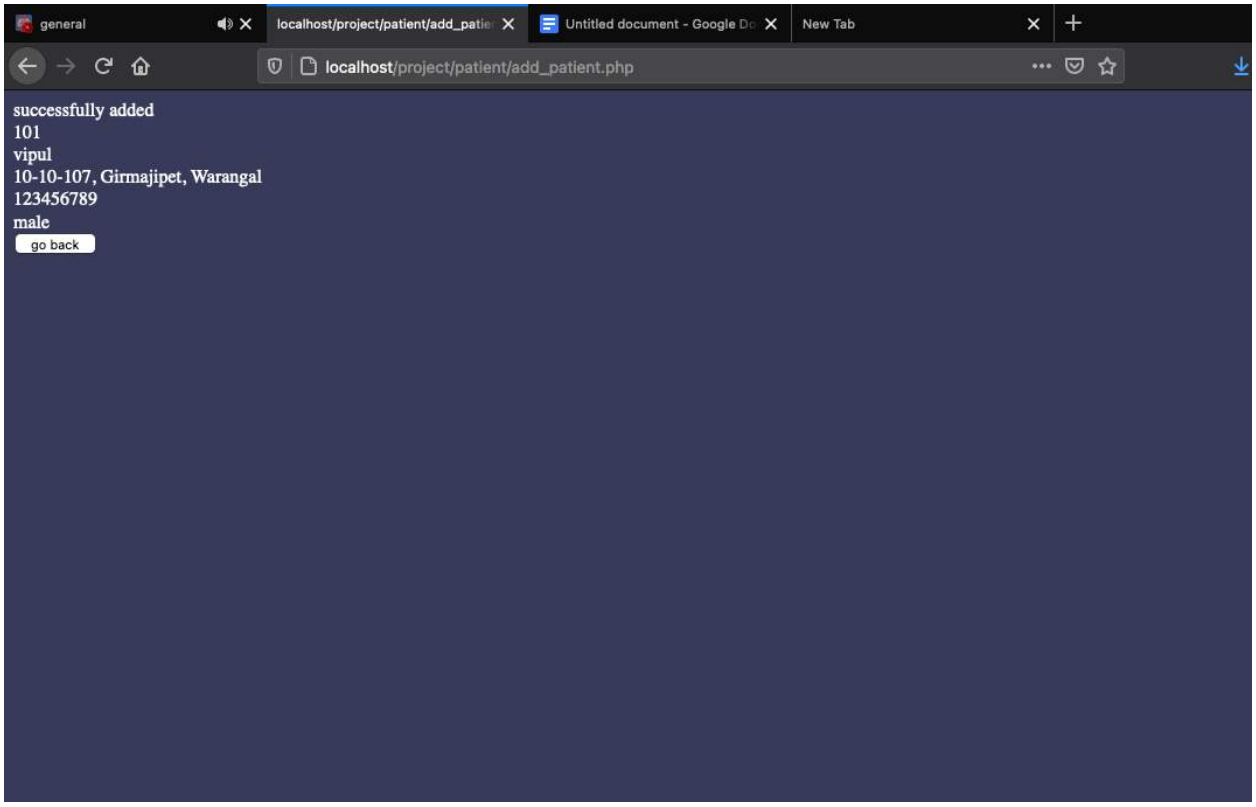
WEB APP

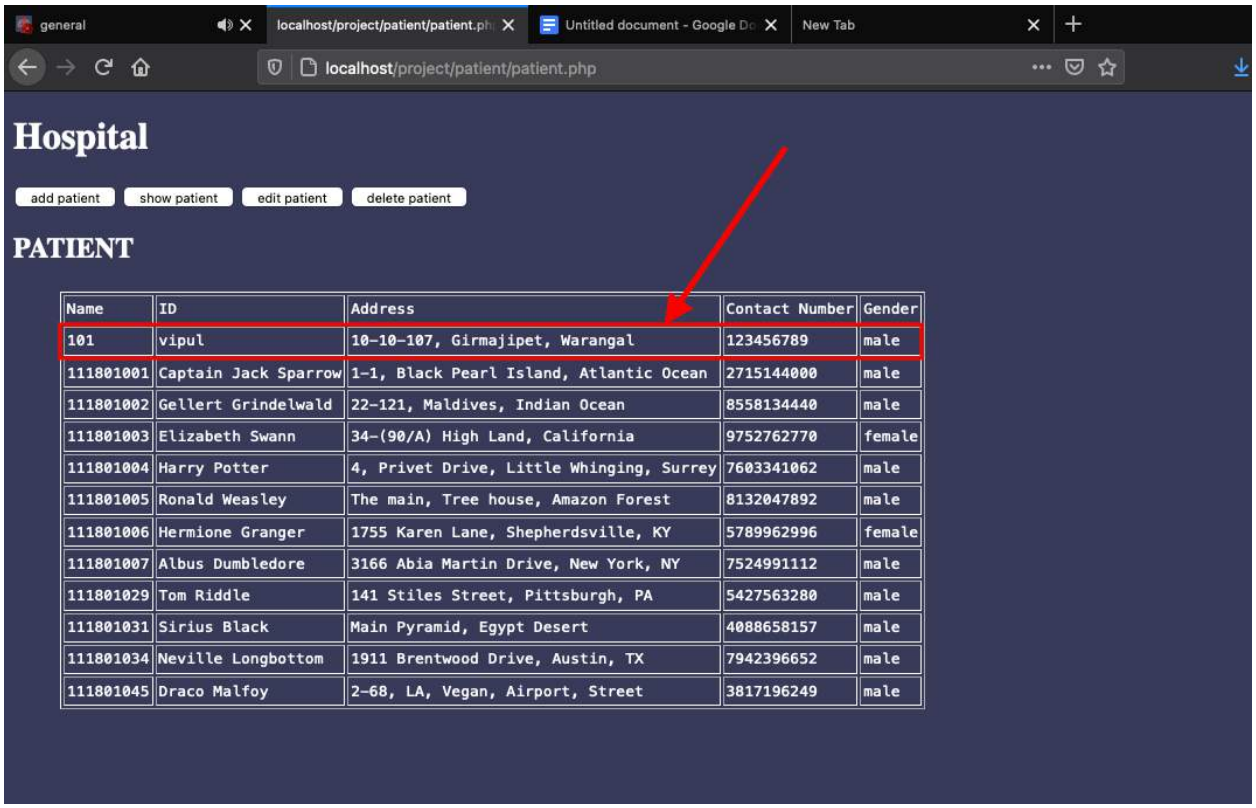
1. Show patient functionality



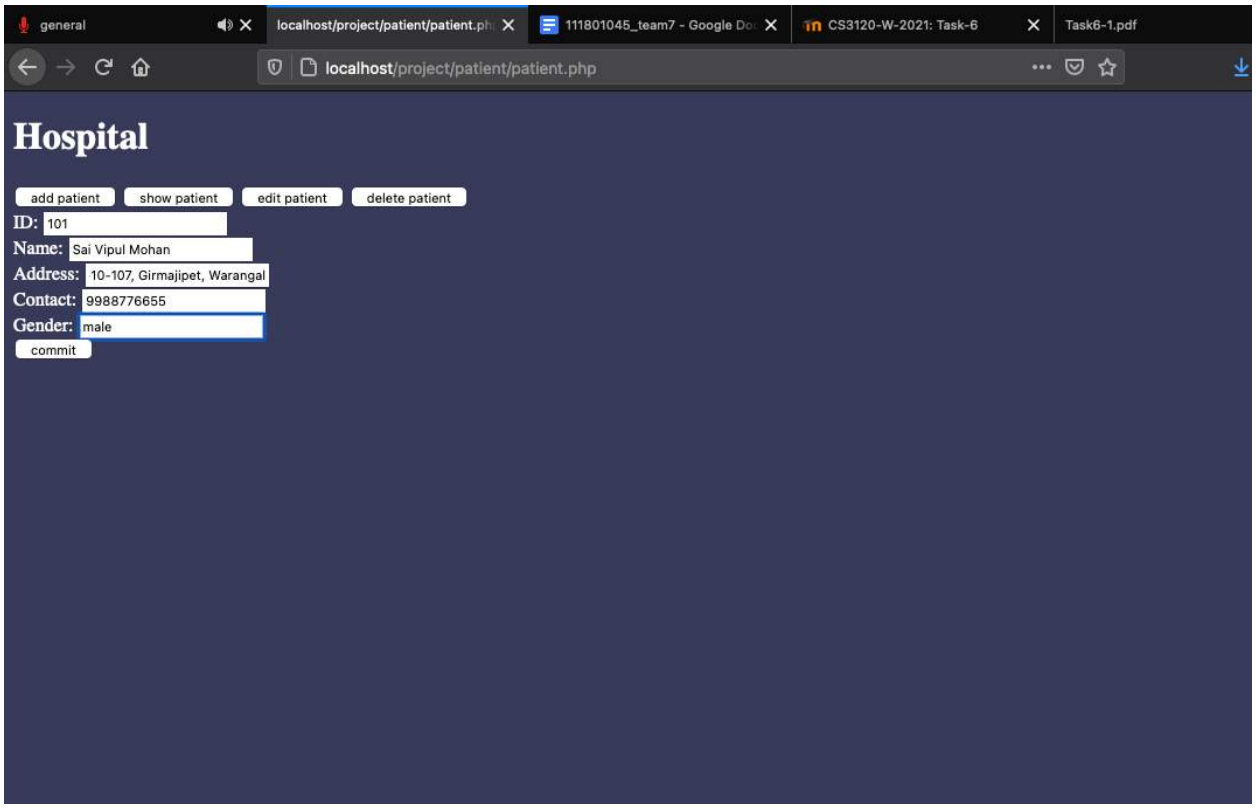
2. Add patient functionality

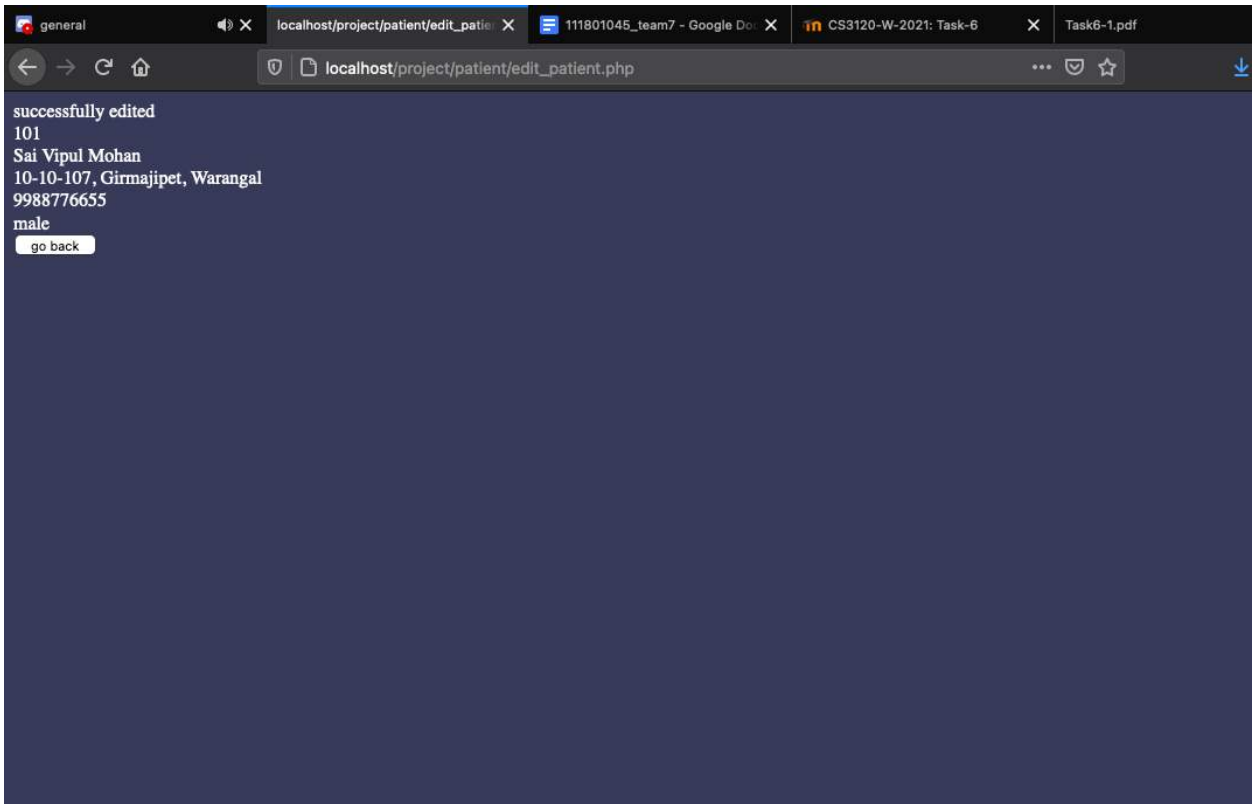


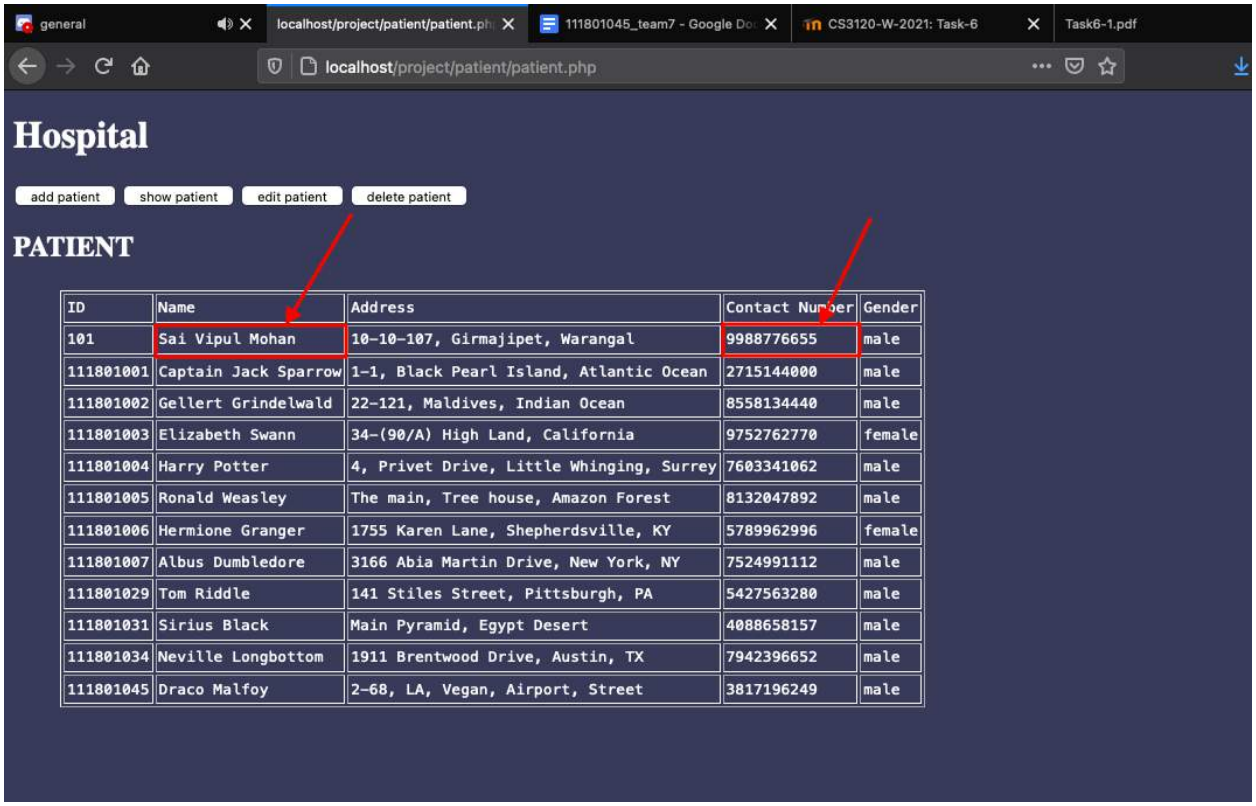

general
localhost/project/patient/add_patie
Untitled document - Google Do
New Tab
localhost/project/patient/add_patient.php
successfully added
101
vipul
10-10-107, Girmajipet, Warangal
123456789
male
go back


general
localhost/project/patient/patient.ph
Untitled document - Google Do
New Tab
localhost/project/patient/patient.php
Hospital
add patient
show patient
edit patient
delete patient
PATIENT
Name | ID | Address | Contact Number | Gender
101 | vipul | 10-10-107, Girmajipet, Warangal | 123456789 | male
111801001 | Captain Jack Sparrow | 1-1, Black Pearl Island, Atlantic Ocean | 2715144000 | male
111801002 | Gellert Grindelwald | 22-121, Maldives, Indian Ocean | 8558134440 | male
111801003 | Elizabeth Swann | 34-(90/A) High Land, California | 9752762770 | female
111801004 | Harry Potter | 4, Privet Drive, Little Whinging, Surrey | 7603341062 | male
111801005 | Ronald Weasley | The main, Tree house, Amazon Forest | 8132047892 | male
111801006 | Hermione Granger | 1755 Karen Lane, Shepherdsville, KY | 5789962996 | female
111801007 | Albus Dumbledore | 3166 Abia Martin Drive, New York, NY | 7524991112 | male
111801029 | Tom Riddle | 141 Stiles Street, Pittsburgh, PA | 5427563280 | male
111801031 | Sirius Black | Main Pyramid, Egypt Desert | 4088658157 | male
111801034 | Neville Longbottom | 1911 Brentwood Drive, Austin, TX | 7942396652 | male
111801045 | Draco Malfoy | 2-68, LA, Vegan, Airport, Street | 3817196249 | male

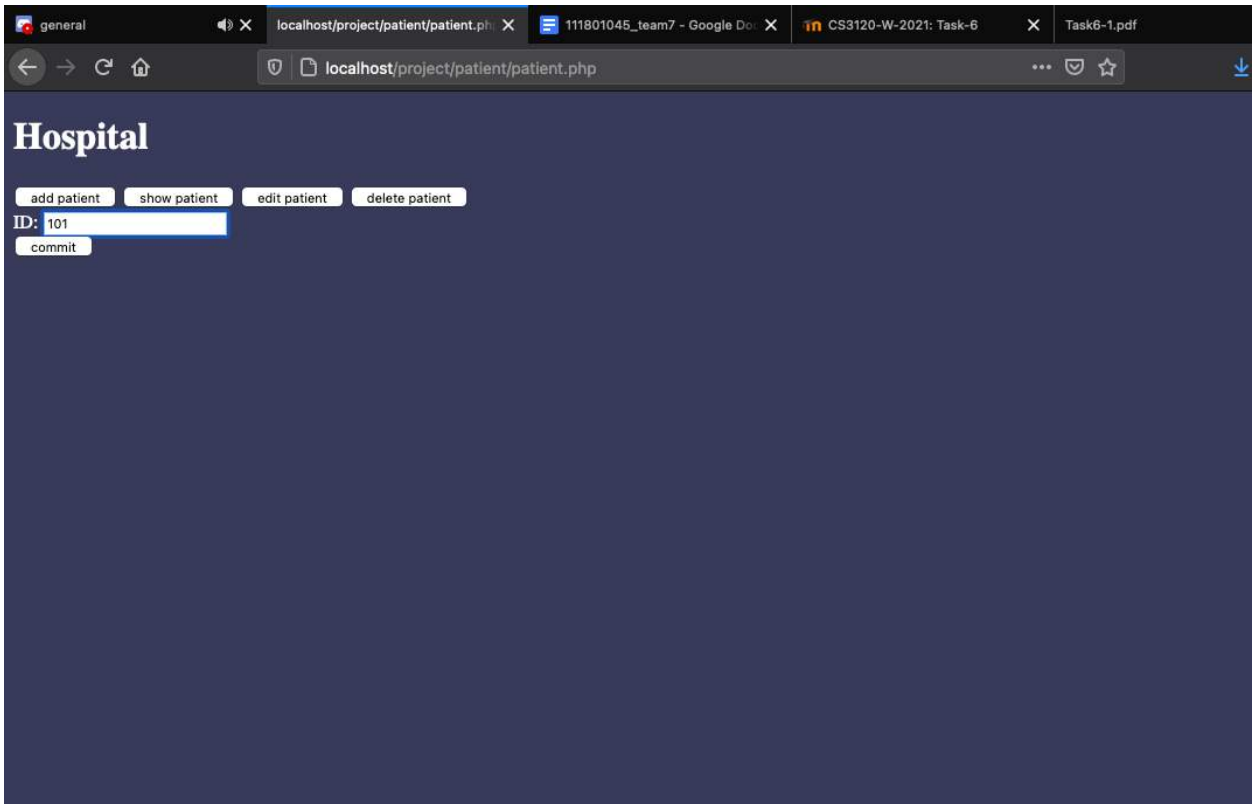
3. Edit patient functionality

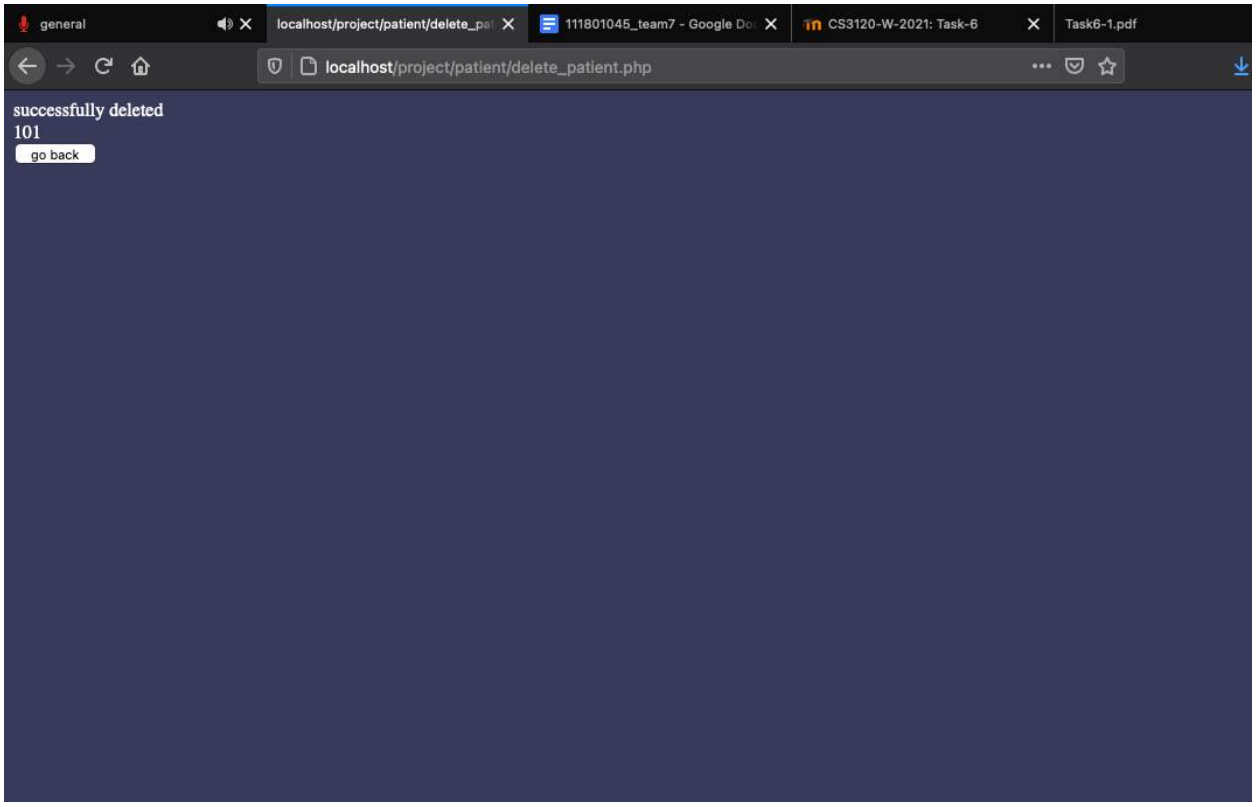


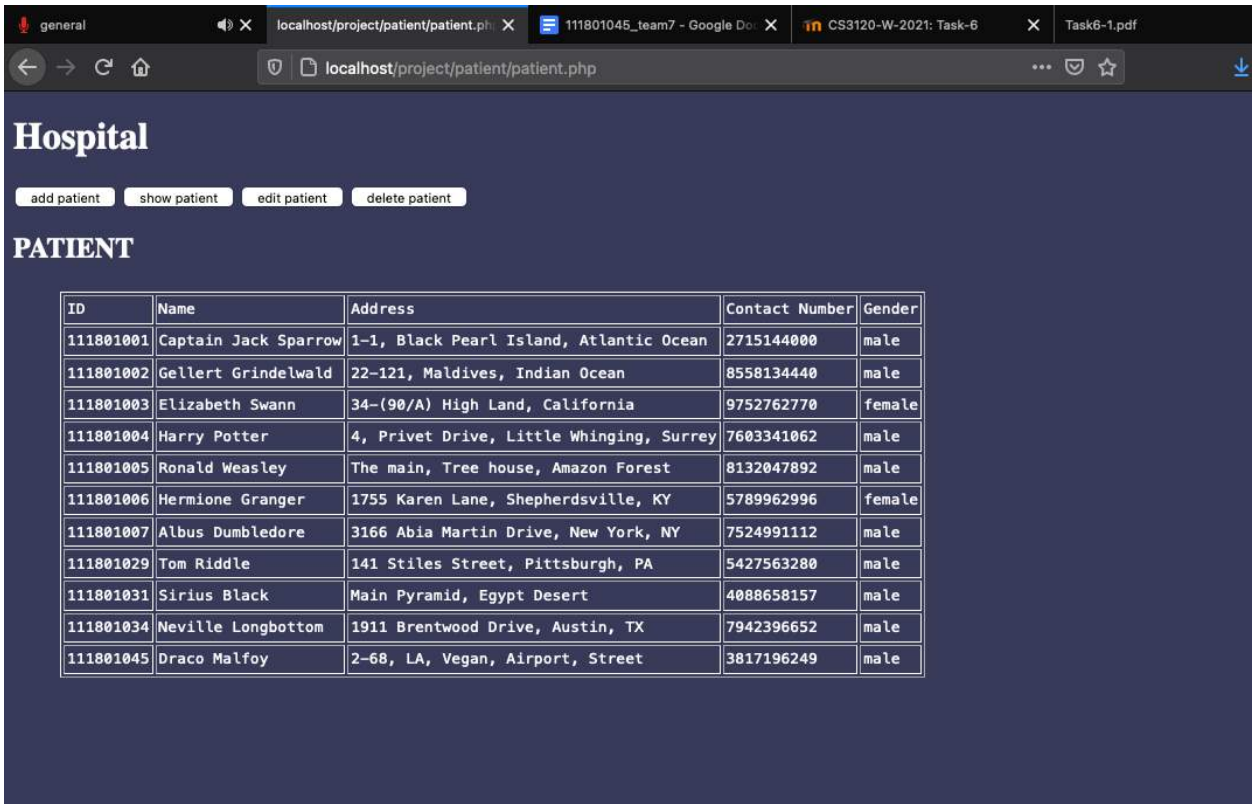




4. Delete patient functionality



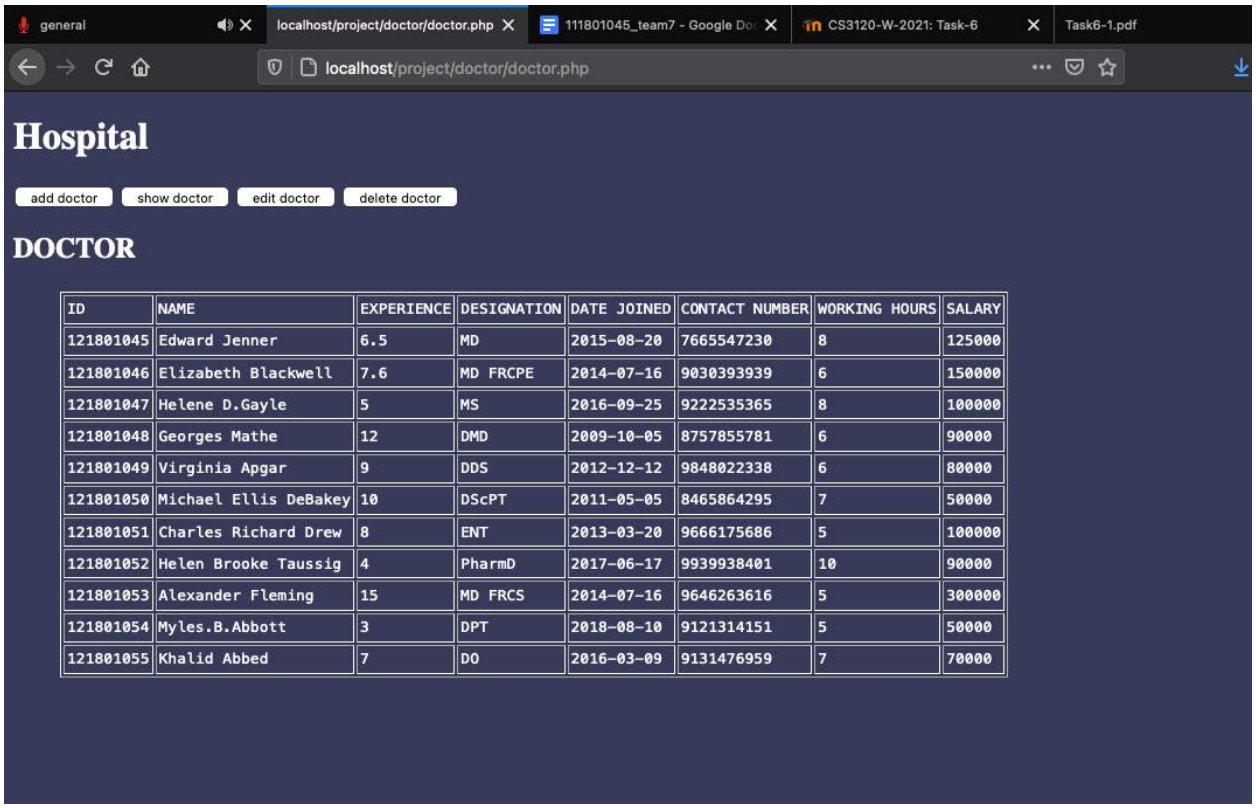




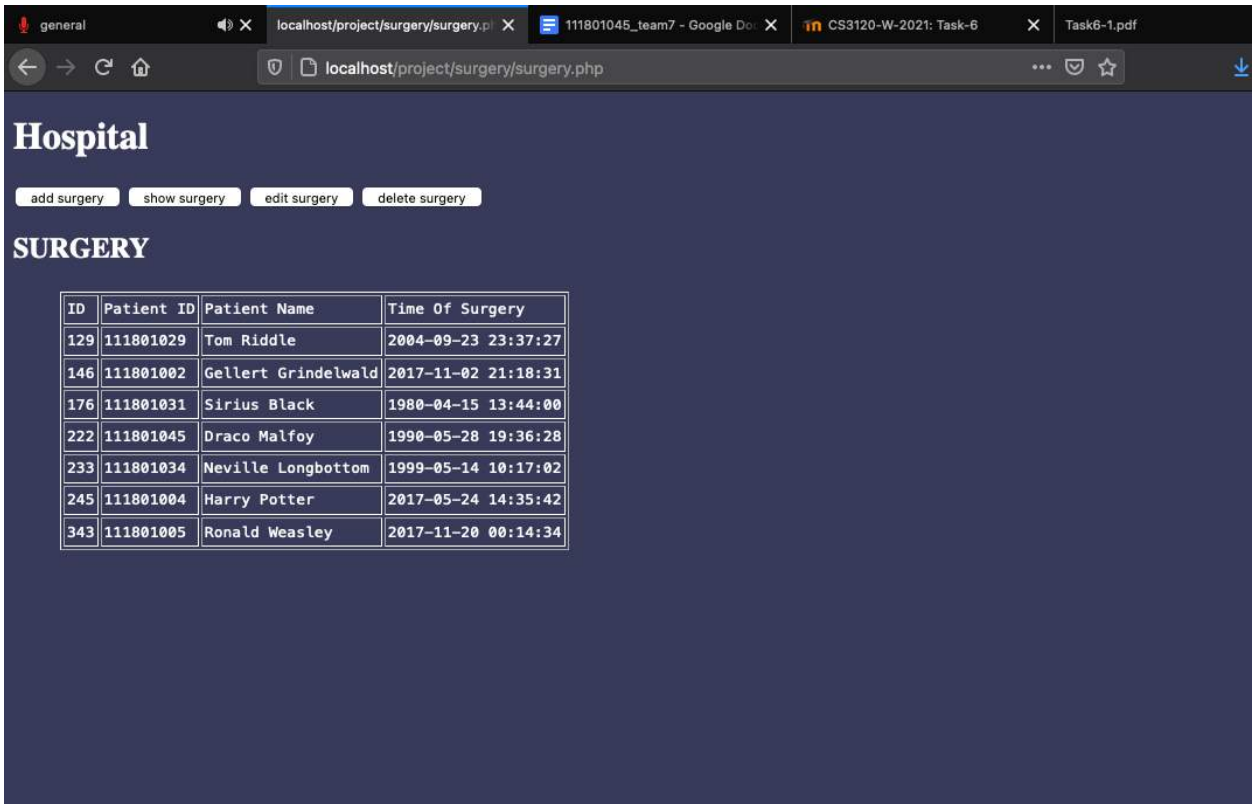
ID	Name	Address	Contact Number	Gender
111801001	Captain Jack Sparrow	1-1, Black Pearl Island, Atlantic Ocean	2715144000	male
111801002	Gellert Grindelwald	22-121, Maldives, Indian Ocean	8558134440	male
111801003	Elizabeth Swann	34-(90/A) High Land, California	9752762770	female
111801004	Harry Potter	4, Privet Drive, Little Whinging, Surrey	7603341062	male
111801005	Ronald Weasley	The main, Tree house, Amazon Forest	8132047892	male
111801006	Hermione Granger	1755 Karen Lane, Shepherdsville, KY	5789962996	female
111801007	Albus Dumbledore	3166 Abia Martin Drive, New York, NY	7524991112	male
111801029	Tom Riddle	141 Stiles Street, Pittsburgh, PA	5427563280	male
111801031	Sirius Black	Main Pyramid, Egypt Desert	4088658157	male
111801034	Neville Longbottom	1911 Brentwood Drive, Austin, TX	7942396652	male
111801045	Draco Malfoy	2-68, LA, Vegan, Airport, Street	3817196249	male

Similar functionality for all the subsequent pages mentioned below

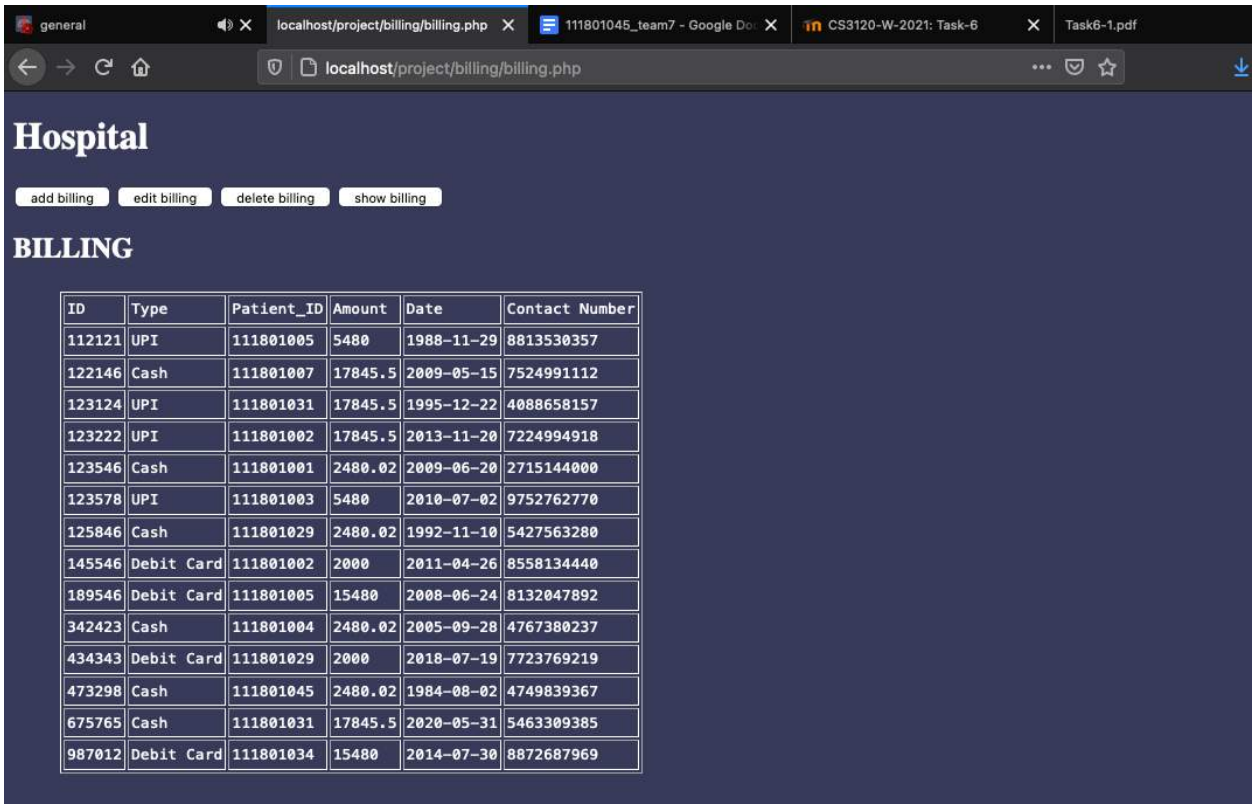
Doctor page



Surgery page

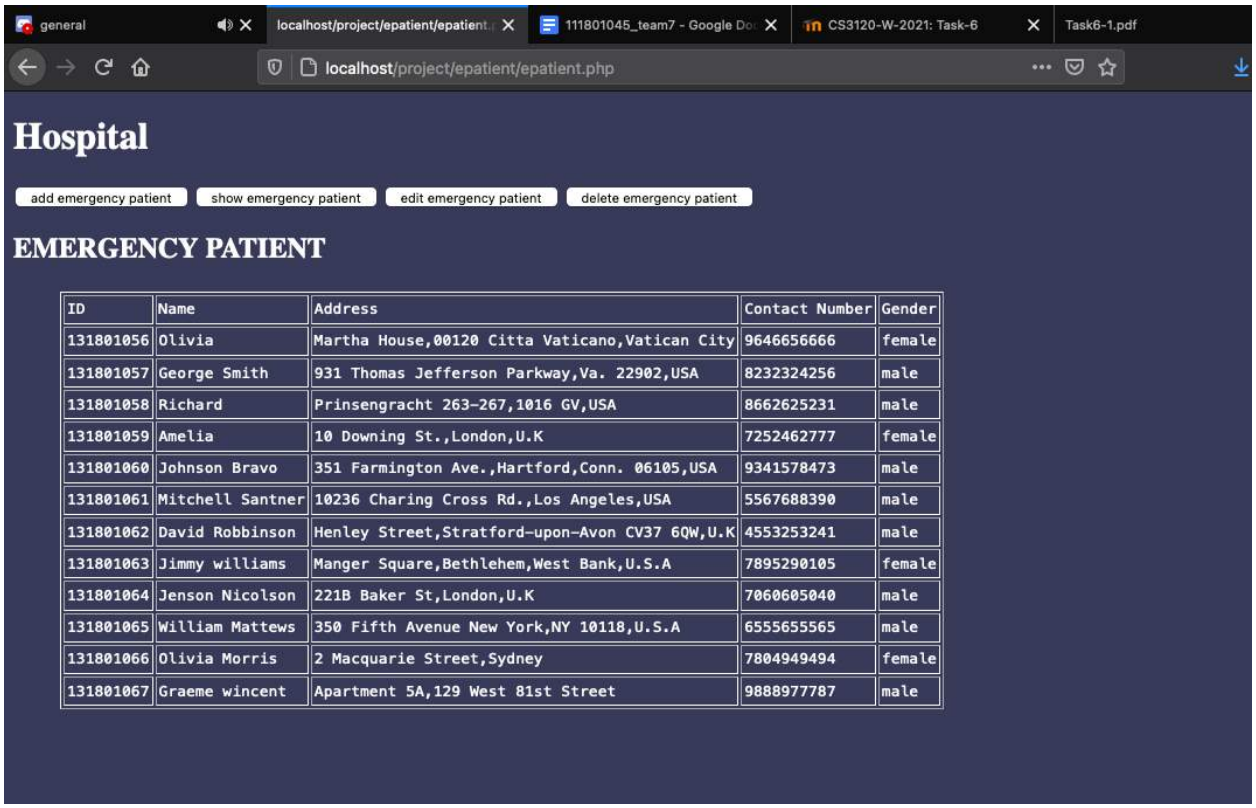


Billing page



ID	Type	Patient_ID	Amount	Date	Contact Number
112121	UPI	111801005	5480	1988-11-29	8813530357
122146	Cash	111801007	17845.5	2009-05-15	7524991112
123124	UPI	111801031	17845.5	1995-12-22	4088658157
123222	UPI	111801002	17845.5	2013-11-20	7224994918
123546	Cash	111801001	2480.02	2009-06-20	2715144000
123578	UPI	111801003	5480	2010-07-02	9752762770
125846	Cash	111801029	2480.02	1992-11-10	5427563280
145546	Debit Card	111801002	2000	2011-04-26	8558134440
189546	Debit Card	111801005	15480	2008-06-24	8132047892
342423	Cash	111801004	2480.02	2005-09-28	4767380237
434343	Debit Card	111801029	2000	2018-07-19	7723769219
473298	Cash	111801045	2480.02	1984-08-02	4749839367
675765	Cash	111801031	17845.5	2020-05-31	5463309385
987012	Debit Card	111801034	15480	2014-07-30	8872687969

Emergency patient page



ID	Name	Address	Contact Number	Gender
131801056	Olivia	Martha House,00120 Citta Vaticano,Vatican City	9646656666	female
131801057	George Smith	931 Thomas Jefferson Parkway,Va. 22902,USA	8232324256	male
131801058	Richard	Prinsengracht 263-267,1016 GV,USA	8662625231	male
131801059	Amelia	10 Downing St.,London,U.K	7252462777	female
131801060	Johnson Bravo	351 Farmington Ave.,Hartford,Conn. 06105,USA	9341578473	male
131801061	Mitchell Santner	10236 Charing Cross Rd.,Los Angeles,USA	5567688390	male
131801062	David Robbinson	Henley Street,Stratford-upon-Avon CV37 6QW,U.K	4553253241	male
131801063	Jimmy williams	Manger Square,Bethlehem,West Bank,U.S.A	7895290105	female
131801064	Jenson Nicolson	221B Baker St,London,U.K	7060605040	male
131801065	William Mattews	350 Fifth Avenue New York,NY 10118,U.S.A	6555655565	male
131801066	Olivia Morris	2 Macquarie Street,Sydney	7804949494	female
131801067	Graeme wincent	Apartment 5A,129 West 81st Street	9888977787	male

Technologies
Mariadb Database
Apache Server
PHP server side language
HTML client side language
(LAMP stack)

Interaction
Apache server is used to serve the php pages.
PHP is used to interact with the database
HTML is used to take input from user and respond accordingly
HTML requests from PHP, PHP fetches from database and serves to HTML
HTML displays the data to the user.

Challenges
Handling edge cases where data entered by the user does not obey integrity constraints of the database.
Handling editing and deleting from database.

APPENDIX

Schema Files

Database Structure Creation file
Database Data Insertion file
Database Structure without data Backup file
Database Structure with data Backup file

Backup Creation

```
mysqldump -u root -p --no-data hospital >
backup_hospital_structure.sql
mysqldump -u root -p hospital > backup_hospital_data.sql
```

Backup Loading

```
mysql -u root -p hospital < backup_hospital_structure.sql
mysql -u root -p hospital < backup_hospital_data.sql
```

Views Files

Views in Database along with justifications of the view