

HOSPITAL MANAGEMENT SYSTEM

Name: V. Sai Vipul Mohan (111801045)

Team Number: 7

Team Details:

Vinay Kumar P (111801034)

Pavan Karthik P (111801031)

Gurunadh P (111801029)

TABLE OF CONTENTS

TABLE OF CONTENTS	2
REQUIREMENT SPECIFICATIONS	3
Introduction	3
Working	3
Requirement	4
ENTITY RELATIONSHIP DIAGRAM	6
SCHEMA DIAGRAM	7
SCHEMA	8
KEY AREAS OF SCHEMA	16
VIEWS	21
FUNCTIONS	28
PROCEDURES	31
WEB APP	34
CONTRIBUTION TO THE PROJECT	42
APPENDIX	43
Schema Files	43
Views Files	43

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, 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.

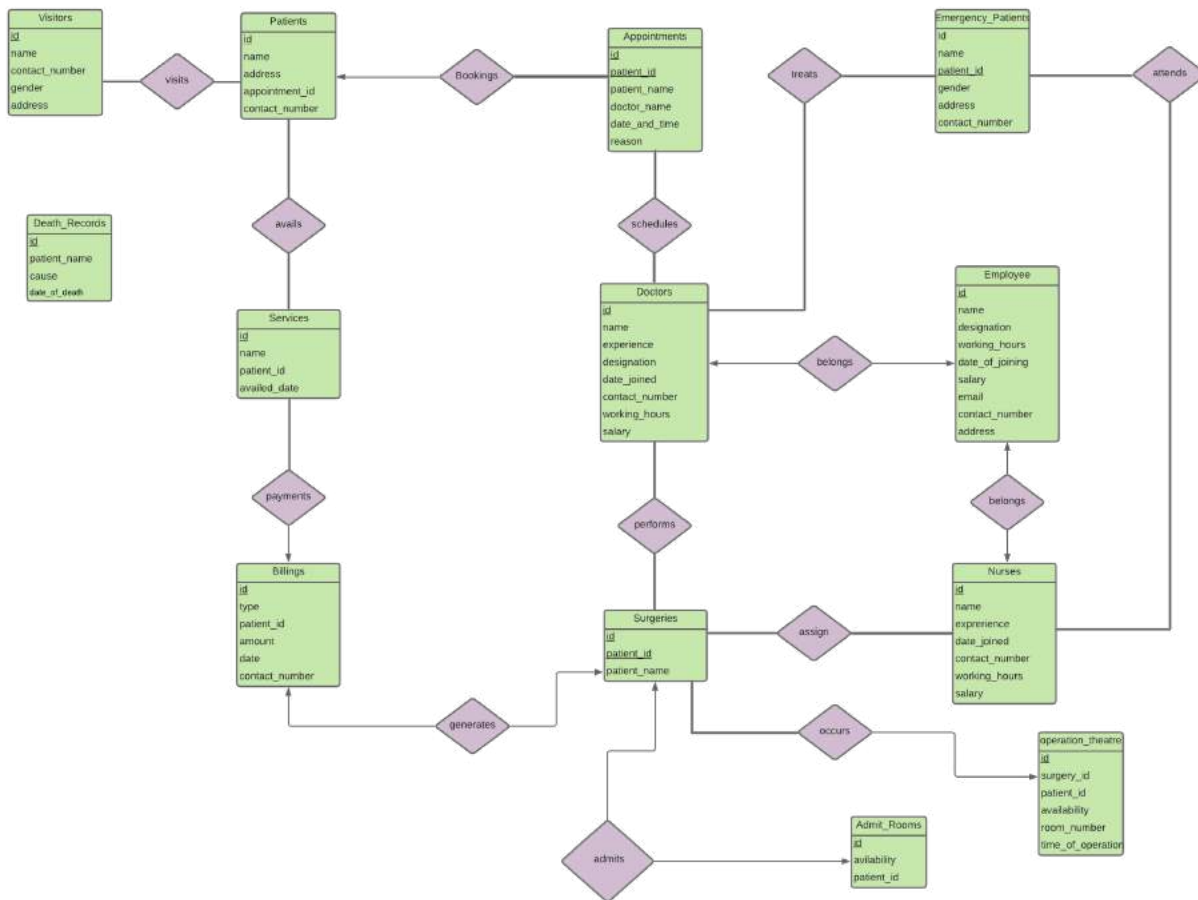
Employees' data is also to be stored, to provide them their salary on time and keep track of their information for the times of any mishap.

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.

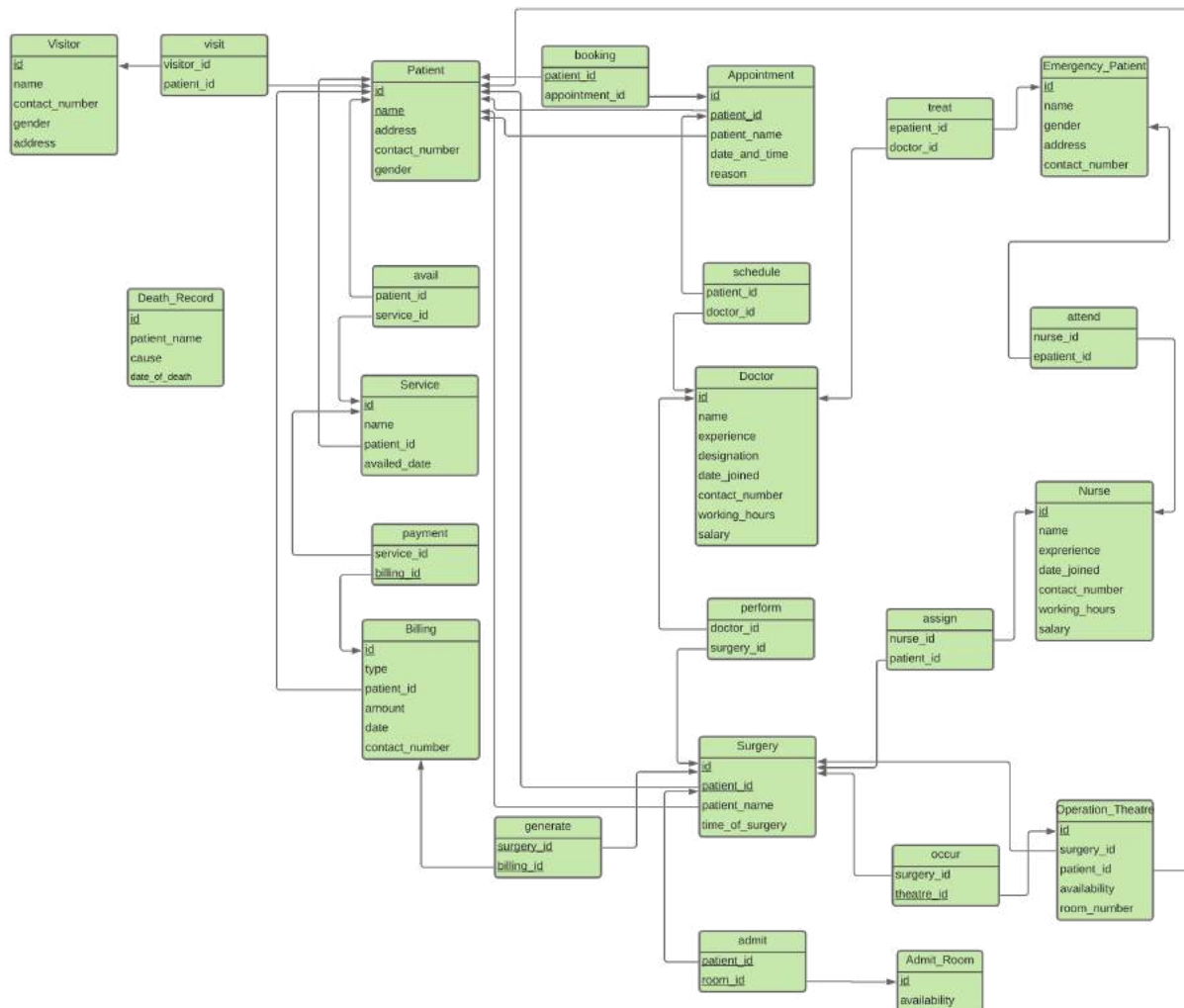
Users

There are two user levels. Admin users can perform all kinds of activities.

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)
```

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)

- 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)

- 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]> select * from patient_appointment_service;
```

patient_id	patient_name	appointment_reason	appointment_date_and_time	service_name	service_availed_date
111801001	Captain Jack Sparrow	Stomach Pain	2021-05-06 00:00:00	Blood test	2009-06-20 00:00:00
111801002	Gellert Grindelwald	Stomach Pain	2021-05-06 00:00:00	HRCT test	2011-04-26 00:00:00
111801003	Elizabeth Swann	Head ache	2021-05-06 00:00:00	X-Ray test	2010-07-02 00:00:00
111801004	Harry Potter	Stomach Pain	2021-05-06 00:00:00	NULL	NULL
111801005	Ronald Weasley	Head ache	2021-05-06 00:00:00	ENT	2008-06-24 00:00:00
111801006	Hermione Granger	Stomach Pain	2021-05-06 00:00:00	NULL	NULL
111801007	Albus Dumbledore	Head ache	2021-05-06 00:00:00	Medical Checkup	2009-05-15 00:00:00
111801029	Tom Riddle	Stomach Pain	2021-05-06 00:00:00	Blood test	1992-11-10 00:00:00
111801031	Sirius Black	Head ache	2021-05-06 00:00:00	Medical Checkup	1995-12-22 00:00:00
111801034	Neville Longbottom	Head ache	2021-05-06 00:00:00	NULL	NULL
111801045	Draco Malfoy	Head ache	2021-05-06 00:00:00	NULL	NULL

11 rows in set (0.006 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]> select * from billing_service_surgery;
```

billing_id	patient_id	billing_amount	billing_date	service_id	service_name	service_availed_date	surgery_id	time_of_surgery
112121	111801005	5480	1988-11-29 00:00:00	19568	ENT	2008-06-24 00:00:00	343	2017-11-20 00:14:34
122146	111801007	17845.5	2009-05-15 00:00:00	11524	Medical Checkup	2009-05-15 00:00:00	NULL	NULL
123124	111801031	17845.5	1995-12-22 00:00:00	17548	Medical Checkup	1995-12-22 00:00:00	176	1980-04-15 13:44:00
123222	111801002	17845.5	2013-11-20 00:00:00	17954	HRCT test	2011-04-26 00:00:00	146	2017-11-02 21:18:31
123546	111801001	2480.02	2009-06-20 00:00:00	12456	Blood test	2009-06-20 00:00:00	000	2017-11-02 21:18:31
123578	111801003	5480	2010-07-02 00:00:00	17845	X-Ray test	2010-07-02 00:00:00	NULL	NULL
125846	111801029	2480.02	1992-11-10 00:00:00	12948	Blood test	1992-11-10 00:00:00	129	2004-09-23 23:37:27
145546	111801002	2000	2011-04-26 00:00:00	17954	HRCT test	2011-04-26 00:00:00	146	2017-11-02 21:18:31
189546	111801005	15480	2008-06-24 00:00:00	19568	ENT	2008-06-24 00:00:00	343	2017-11-20 00:14:34
342423	111801004	2480.02	2005-09-28 00:00:00	NULL	NULL	NULL	245	2017-05-24 14:35:42
434343	111801029	2000	2018-07-19 00:00:00	12948	Blood test	1992-11-10 00:00:00	129	2004-09-23 23:37:27
473298	111801045	2480.02	1984-08-02 00:00:00	NULL	NULL	NULL	222	1990-05-28 19:36:28
675765	111801031	17845.5	2020-05-31 00:00:00	17548	Medical Checkup	1995-12-22 00:00:00	176	1980-04-15 13:44:00
987012	111801034	15480	2014-07-30 00:00:00	NULL	NULL	NULL	233	1999-05-14 10:17:02

14 rows in set (0.001 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]> select * from patient_epatient;
```

patient_id	patient_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
131801056	Olivia
131801057	George Smith
131801058	Richard
131801059	Amelia
131801060	Johnson Bravo
131801061	Mitchell Santner
131801062	David Robinson
131801063	Jimmy Williams
131801064	Jenson Nicolson
131801065	William Matthews
131801066	Olivia Morris
131801067	Graeme Wincent

23 rows in set (0.001 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]> select * from employee;
```

employee_id	employee_name	working_hours	employee_salary
121801045	Edward Jenner	8	125000
121801046	Elizabeth Blackwell	6	150000
121801047	Helene D.Gayle	8	100000
121801048	Georges Mathe	6	90000
121801049	Virginia Apgar	6	80000
121801050	Michael Ellis DeBakey	7	50000
121801051	Charles Richard Drew	5	100000
121801052	Helen Brooke Taussig	10	90000
121801053	Alexander Fleming	5	300000
121801054	Myles.B.Abbott	5	50000
121801055	Khalid Abbed	7	70000
141801001	Julie Watson	9	20000
141801002	Marie Phillips	7	10000
141801003	Tiffany Morrison	5	8000
141801004	Jamie Rose	4	7000
141801005	Emily Parker	8	15000
141801006	Jamie Thomas	10	25000
141801007	Sophie Jane	5	6000
141801008	Lily Robberts	6	9000
141801009	Jennifer seifert	7	9500
141801010	Stephany Johnson	8	6500

21 rows in set (0.001 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]> select * from nurse_assign_surgery;
```

id	patient_id	patient_name	nurse_name
245	111801004	Harry Potter	Julie Watson
343	111801005	Ronald Weasley	Emily Parker
129	111801029	Tom Riddle	Jamie Rose
176	111801031	Sirius Black	Tiffany Morrison
233	111801034	Neville Longbottom	Sophie Jane

5 rows in set (0.001 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]> select * from surgery_admit_admitroom;
```

id	patient_id	patient_name	room_id	availability
245	111801004	Harry Potter	101	1
343	111801005	Ronald Weasley	102	1
129	111801029	Tom Riddle	103	1
176	111801031	Sirius Black	104	1
222	111801045	Draco Malfoy	105	1
146	111801002	Gellert Grindelwald	106	1

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]> select * from surgery_occur_operationtheatre;
```

id	patient_id	patient_name	operation_theatre_id	availability
343	111801005	Ronald Weasley	002	1
129	111801029	Tom Riddle	003	1
176	111801031	Sirius Black	004	1
233	111801034	Neville Longbottom	005	0

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]> select * from nurse_assign_patient;
```

nurse_id	nurse_name	patient_Id	patient_name
141801001	Julie Watson	111801004	Harry Potter
141801005	Emily Parker	111801005	Ronald Weasley
141801004	Jamie Rose	111801029	Tom Riddle
141801003	Tiffany Morrison	111801031	Sirius Black
141801007	Sophie Jane	111801034	Neville Longbottom

5 rows in set (0.001 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]> select * from doctor_schedule_patient;
```

Doctor_id	Doctor_name	designation	patient_id	patient_name
121801045	Edward Jenner	MD	111801001	Captain Jack Sparrow
121801046	Elizabeth Blackwell	MD FRCPE	111801002	Gellert Grindelwald
121801047	Helene D.Gayle	MS	111801003	Elizabeth Swann
121801048	Georges Mathe	DMD	111801004	Harry Potter
121801049	Virginia Apgar	DDS	111801005	Ronald Weasley
121801050	Michael Ellis DeBakey	DScPT	111801006	Hermione Granger
121801051	Charles Richard Drew	ENT	111801007	Albus Dumbledore
121801052	Helen Brooke Taussig	PharmD	111801029	Tom Riddle
121801053	Alexander Fleming	MD FRCS	111801031	Sirius Black
121801054	Myles.B.Abbott	DPT	111801034	Neville Longbottom
121801055	Khalid Abbed	DO	111801045	Draco Malfoy

11 rows in set (0.001 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]> select * from visitor_visit_patient;
```

vistor_id	visitor_name	vistor_number	patient_id	patient_name
v111801001	Will Turner	3715144000	111801001	Captain Jack Sparrow
v111801001	Will Turner	3715144000	111801002	Gellert Grindelwald
v111801002	James Norry	9558134440	111801001	Captain Jack Sparrow
v111801003	Hector	8752762770	111801003	Elizabeth Swann
v111801003	Hector	8752762770	111801004	Harry Potter
v111801004	Ragetti	8603341062	111801003	Elizabeth Swann
v111801005	Ana Maria	9132047892	111801005	Ronald Weasley
v111801005	Ana Maria	9132047892	111801006	Hermione Granger
v111801006	Joshamee	6789962996	111801005	Ronald Weasley
v111801007	Marty	8524991112	111801029	Tom Riddle
v111801008	Mull roy	6427563280	111801031	Sirius Black
v111801009	Murtogg	5088658157	111801034	Neville Longbottom
v111801010	Lieutenant	8942396652	111801045	Draco Malfoy

13 rows in set (0.001 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 patient_id in this view to access the list of visitors directly in a simple query

11. View Name: doctor_epatient_nurse

```
MariaDB [hospital]> select * from doctor_epatient_nurse;
```

doctor_name	doctor_id	e_patient_id	e_patient_name	nurse_id	nurse_name
Edward Jenner	121801045	131801060	Johnson Bravo	141801003	Tiffany Morrison
Helene D.Gayle	121801047	131801056	Olivia	141801001	Julie Watson
Virginia Apgar	121801049	131801057	George Smith	141801010	Stephany Johnson
Edward Jenner	121801045	131801066	Olivia Morris	141801005	Emily Parker
Charles Richard Drew	121801051	131801064	Jenson Nicolson	141801003	Tiffany Morrison
Virginia Apgar	121801049	131801063	Jimmy Williams	141801001	Julie Watson
Edward Jenner	121801045	131801061	Mitchell Santner	141801002	Marie Phillips

7 rows in set (0.010 sec)

Justification:

This is a useful view, just in case we want to have a page listing all the emergency patients 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]> select * from doctor_surgery_nurse;
```

doctor_id	doctor_name	surgery_id	patient_id	patient_name	nurse_id	nurse_name
121801045	Edward Jenner	343	111801005	Ronald Weasley	141801005	Emily Parker
121801046	Elizabeth Blackwell	129	111801029	Tom Riddle	141801004	Jamie Rose
121801051	Charles Richard Drew	176	111801031	Sirius Black	141801003	Tiffany Morrison
121801047	Helene D.Gayle	245	111801004	Harry Potter	141801001	Julie Watson
121801048	Georges Mathe	233	111801034	Neville Longbottom	141801007	Sophie Jane

5 rows in set (0.001 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]> select * from doctor_appointments;
```

doctor_id	doctor_name	appointment_id	patient_name	patient_id	reason
121801045	Edward Jenner	a111801001	Captain Jack Sparrow	111801001	Stomach Pain
121801046	Elizabeth Blackwell	a111801003	Gellert Grindelwald	111801002	Stomach Pain
121801047	Helene D.Gayle	a111801004	Elizabeth Swann	111801003	Head ache
121801048	Georges Mathe	a111801005	Harry Potter	111801004	Stomach Pain
121801049	Virginia Appgar	a111801006	Ronald Weasley	111801005	Head ache
121801050	Michael Ellis DeBakey	a111801007	Hermione Granger	111801006	Stomach Pain
121801051	Charles Richard Drew	a111801008	Albus Dumbledore	111801007	Head ache
121801052	Helen Brooke Taussig	a111801009	Tom Riddle	111801029	Stomach Pain
121801053	Alexander Fleming	a111801010	Sirius Black	111801031	Head ache
121801054	Myles.B.Abbott	a111801011	Neville Longbottom	111801034	Head ache
121801055	Khalid Abbed	a111801012	Draco Malfoy	111801045	Head ache

11 rows in set (0.001 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]> select * from patient_service;
```

patient_id	patient_name	service_id	service_name	date_availed
111801001	Captain Jack Sparrow	12456	Blood test	2009-06-20 00:00:00
111801003	Elizabeth Swann	17845	X-Ray test	2010-07-02 00:00:00
111801002	Gellert Grindelwald	17954	HRCT test	2011-04-26 00:00:00
111801007	Albus Dumbledore	11524	Medical Checkup	2009-05-15 00:00:00
111801005	Ronald Weasley	19568	ENT	2008-06-24 00:00:00
111801029	Tom Riddle	12948	Blood test	1992-11-10 00:00:00
111801031	Sirius Black	17548	Medical Checkup	1995-12-22 00:00:00

7 rows in set (0.001 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. Function Name: 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 earned by the doctors in the hospital.

2. Function Name: 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 the hospital is doing in that particular location.

3. Function Name: 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. Function Name: 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)
```



```

MariaDB [hospital]> SELECT get_employee_name('121801047')#
+-----+
| get_employee_name('121801047') |
+-----+
| Helene D.Gayle                  |
+-----+
1 row in set (0.002 sec)

MariaDB [hospital]> SELECT get_employee_name('141801001')#
+-----+
| get_employee_name('141801001') |
+-----+
| Julie Watson                   |
+-----+
1 row in set (0.001 sec)

MariaDB [hospital]> SELECT get_employee_name('invalid id')#
+-----+
| get_employee_name('invalid id') |
+-----+
| NULL                             |
+-----+
1 row in set (0.001 sec)

MariaDB [hospital]> █

```

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 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]> 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)
```

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 APPLICATION

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

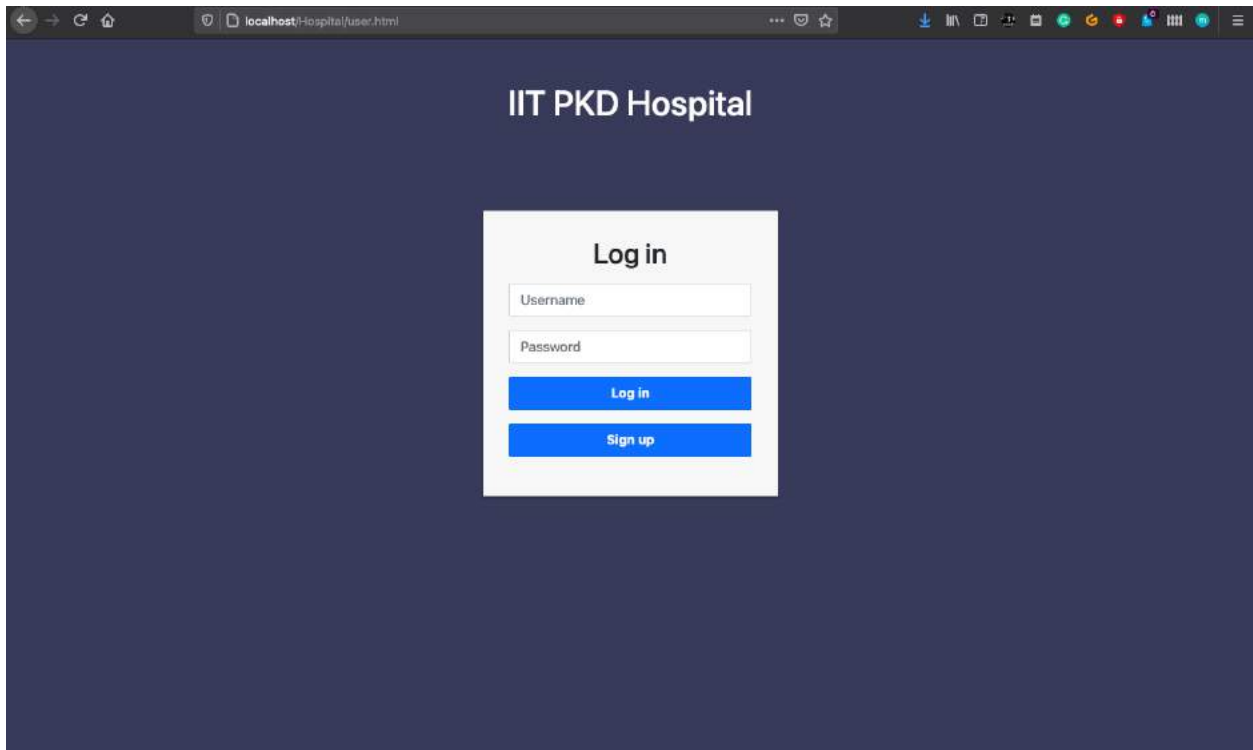
Handling edge cases where data entered by the user does not obey integrity constraints of the database.

Handling editing and deleting from database.

Flow

Flow of the functionality present in the web application is explained in detail in subsequent sections using an example of surgery page.

USER REGISTRATION LOGIN PAGE



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/hospital/user.html'. The page has a dark blue background. At the top center, the text 'IIT PKD Hospital' is displayed in white. Below this, centered, is a white rectangular box containing the login and registration interface. The box is titled 'Log in' in bold. It features two input fields: 'Username' and 'Password'. Below these fields are two blue buttons: 'Log in' and 'Sign up'.

A user must have administration access in the database, or a user must sign up. Only in these cases is a user allowed to enter into the IIT PKD Hospital Web Application.

IIT PKD HOSPITAL HOME PAGE

IIT PKD Hospital

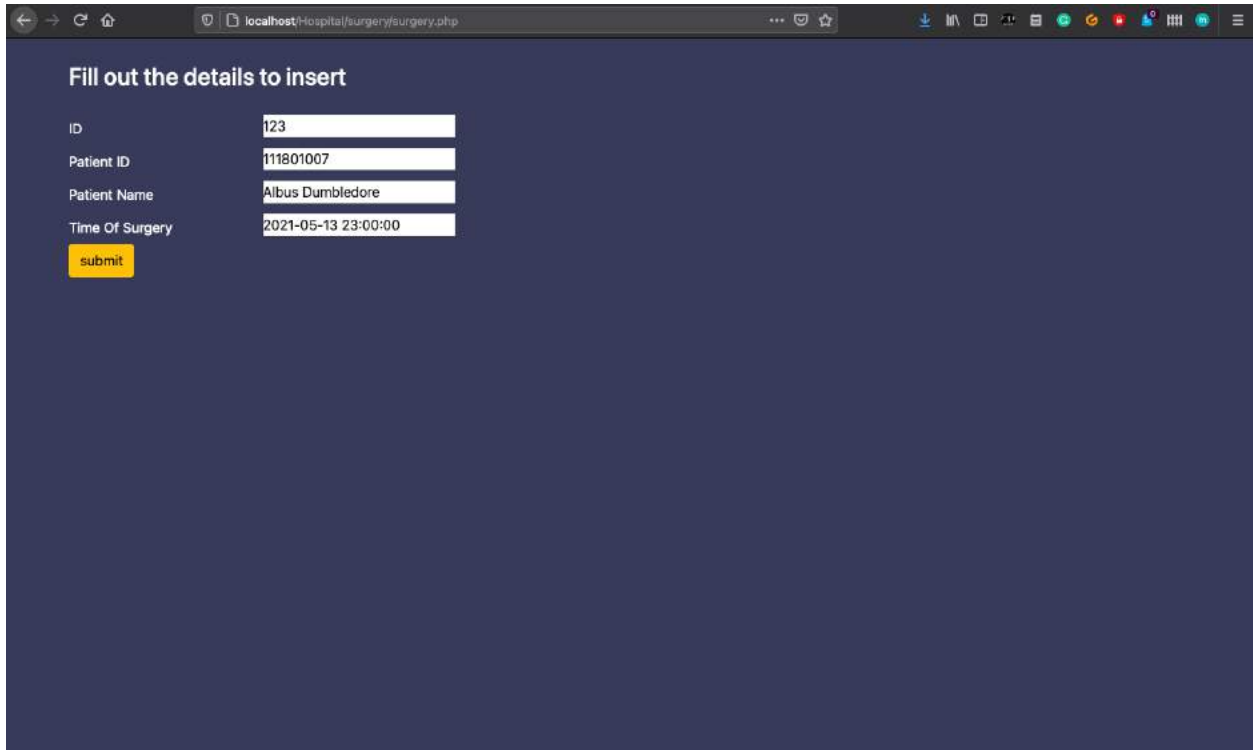
Hospital Management System is majorly to organise Patients, Employees and Services provided. Patients are the customers of the Hospital Management System, every data corresponding to the Patients must be stored and they must be provided with Services or Medication requested. There are three categories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

Hospital Stats	
Patients	11
Employees	21
Surgeries	8
OT's available	1
Revenue Generated So Far	127222

The first landing page after a user enters into the IIT PKD Hospital web application, is the home page. Where details regarding the hospital working can be found. Also, users are constantly updated with the hospital statistics. Users can navigate between different sections of the hospital using the navigation bar present in this page.

SURGERY PAGE-INSERTION

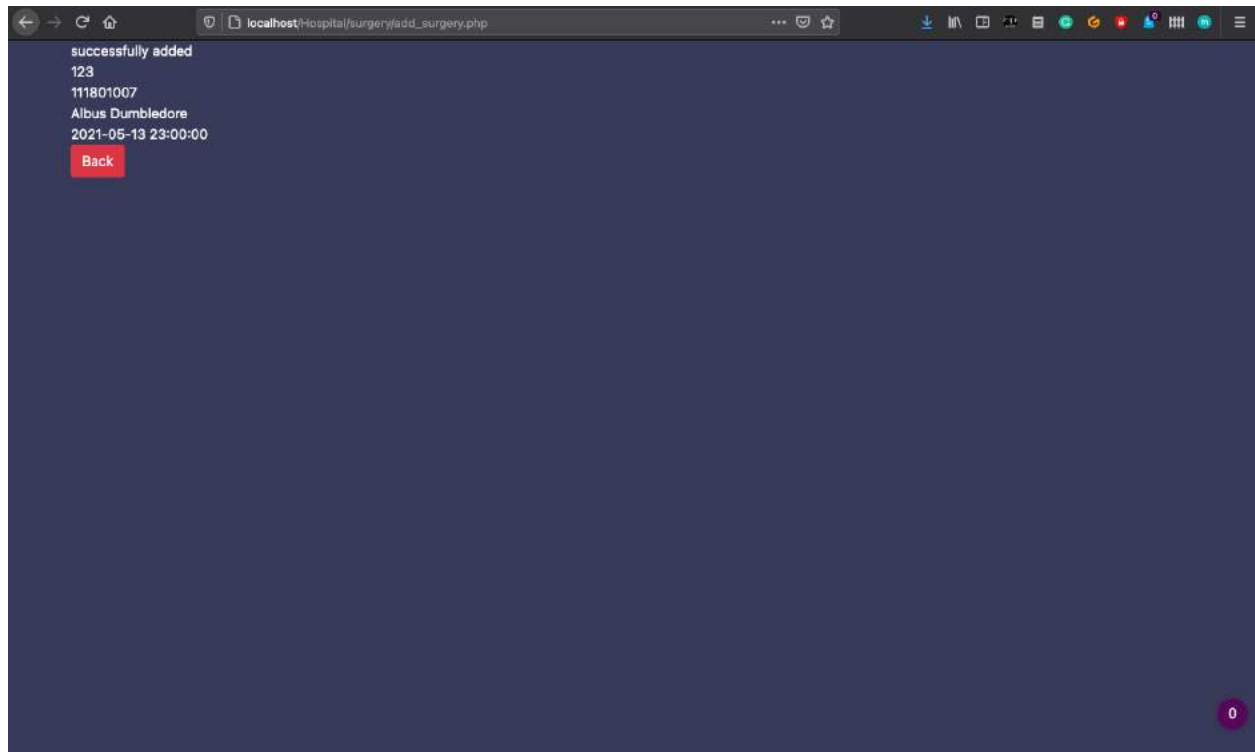


The screenshot shows a web browser window with the address bar displaying 'localhost/hospital/surgery/surgery.php'. The page has a dark blue background and contains a form titled 'Fill out the details to insert'. The form includes four input fields: 'ID' with the value '123', 'Patient ID' with the value '111801007', 'Patient Name' with the value 'Albus Dumbledore', and 'Time Of Surgery' with the value '2021-05-13 23:00:00'. A yellow 'submit' button is located below the input fields.

Fill out the details to insert	
ID	123
Patient ID	111801007
Patient Name	Albus Dumbledore
Time Of Surgery	2021-05-13 23:00:00

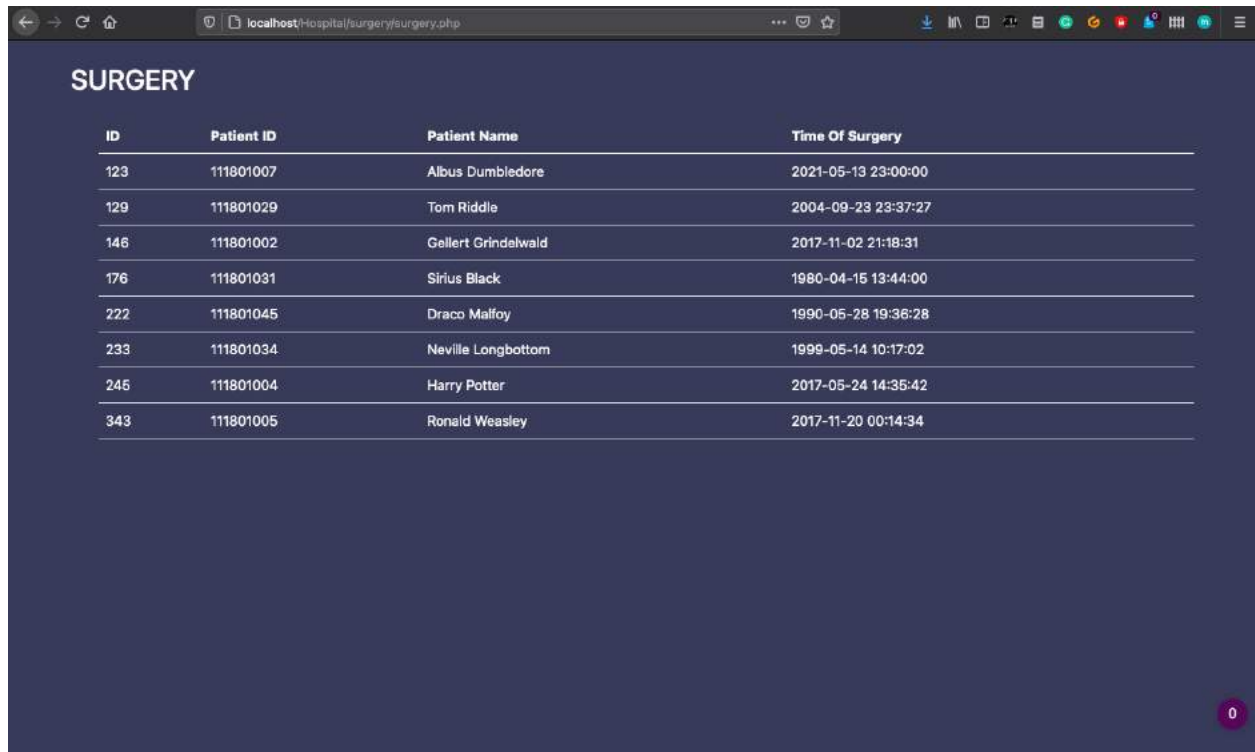
To insert any element into the hospital database, one needs administrative access. And an entry will be added to the database according to the details filled in the given input field. The input fields have empty checks, so empty data will not be entered into the database.

SURGERY PAGE-INSERTION MESSAGE



After the insertion of an entry into the database, if the insertion is successful then a success message is displayed along with the details that are successfully modified. Back button takes the user back to the page from where they landed here(in this case surgery page).

SURGERY PAGE-VIEWING INSERTED DATA

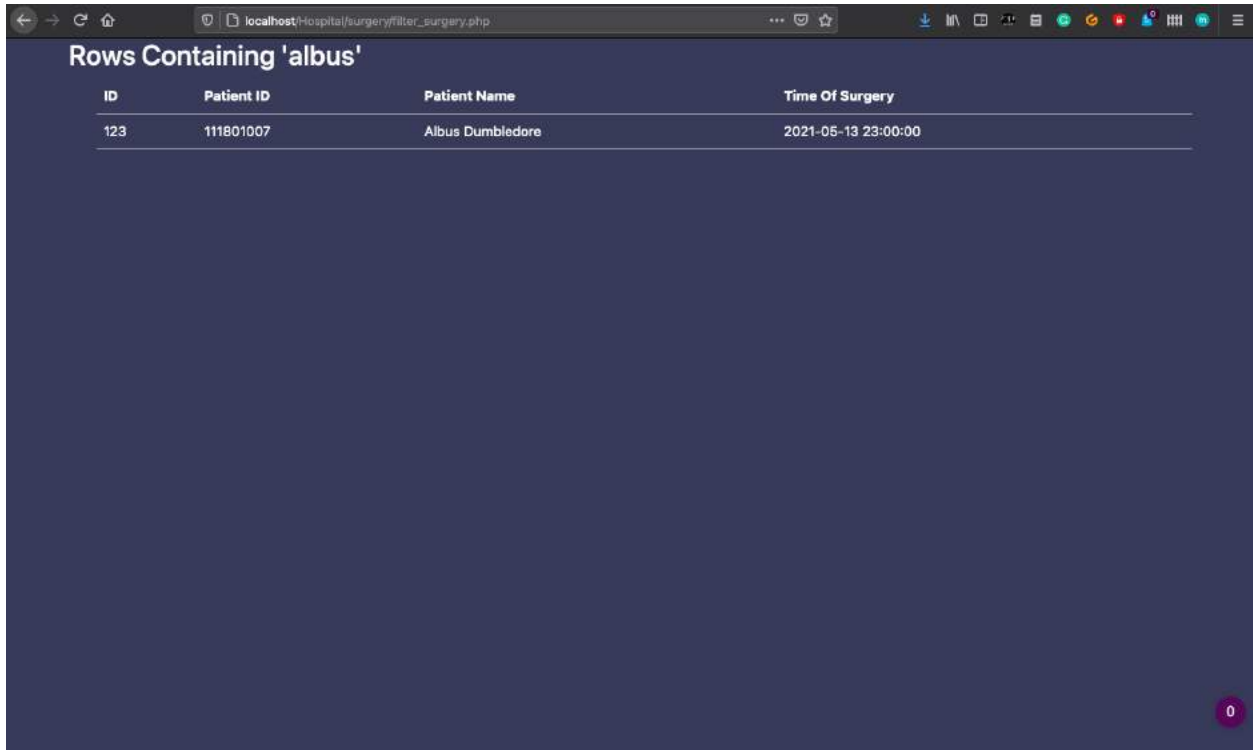


The screenshot shows a web browser window with the address bar displaying 'localhost/hospital/surgery/surgery.php'. The page has a dark blue background and a title 'SURGERY' in white. Below the title is a table with four columns: 'ID', 'Patient ID', 'Patient Name', and 'Time Of Surgery'. The table contains eight rows of data, including characters like Albus Dumbledore, Tom Riddle, and Harry Potter. A small purple circle with the number '0' is visible in the bottom right corner of the page.

ID	Patient ID	Patient Name	Time Of Surgery
123	111801007	Albus Dumbledore	2021-05-13 23:00:00
129	111801029	Tom Riddle	2004-09-23 23:37:27
146	111801002	Gellert Grindelwald	2017-11-02 21:18:31
176	111801031	Sirius Black	1980-04-15 13:44:00
222	111801045	Draco Malfoy	1990-05-28 19:36:28
233	111801034	Neville Longbottom	1999-05-14 10:17:02
246	111801004	Harry Potter	2017-05-24 14:35:42
343	111801005	Ronald Weasley	2017-11-20 00:14:34

Inserted data can be viewed in the show page.

SURGERY PAGE-SEARCHING FOR 'albus'

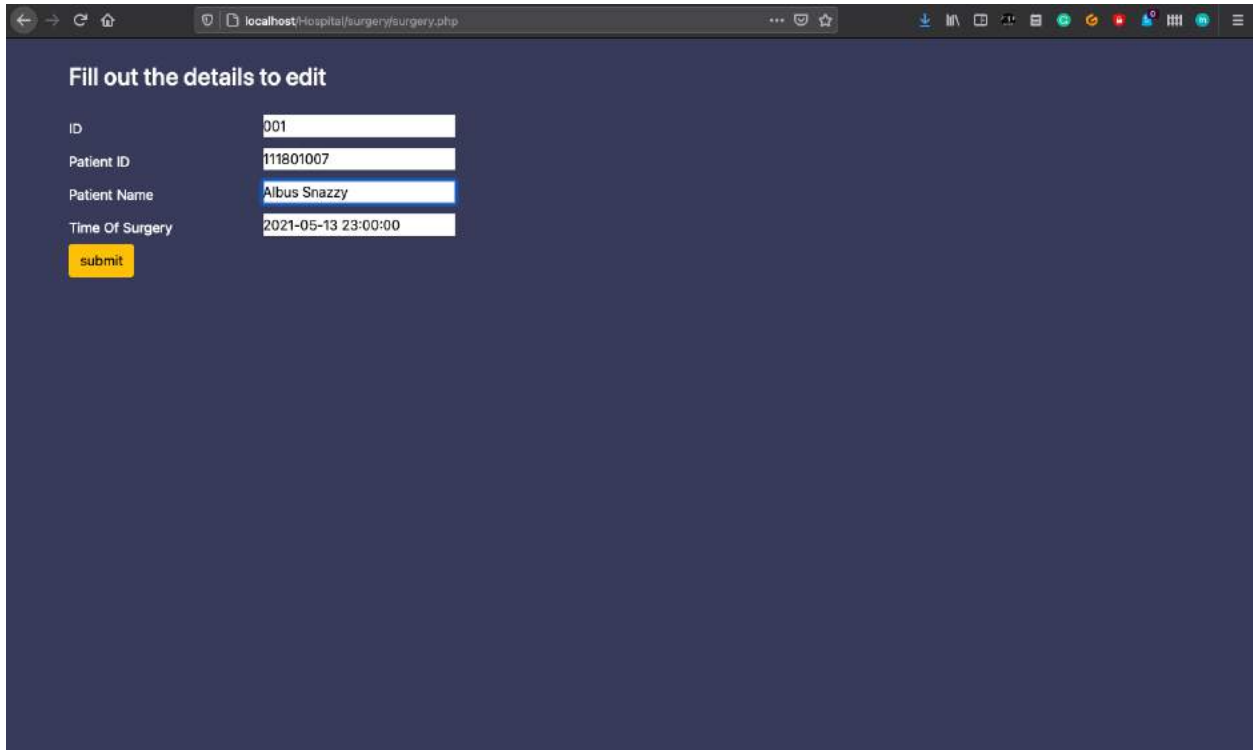


The screenshot shows a web browser window with the address bar displaying 'localhost/hospital/surgery/filter_surgery.php'. The page title is 'Rows Containing 'albus''. Below the title is a table with four columns: 'ID', 'Patient ID', 'Patient Name', and 'Time Of Surgery'. The table contains one row of data. The background of the page is dark blue. In the bottom right corner, there is a small purple circle with the number '0' inside it.

ID	Patient ID	Patient Name	Time Of Surgery
123	111801007	Albus Dumbledore	2021-05-13 23:00:00

Newly inserted data can be searched and the results are shown according to regular expression matching.

SURGERY PAGE-EDITING 'Albus' SECOND NAME



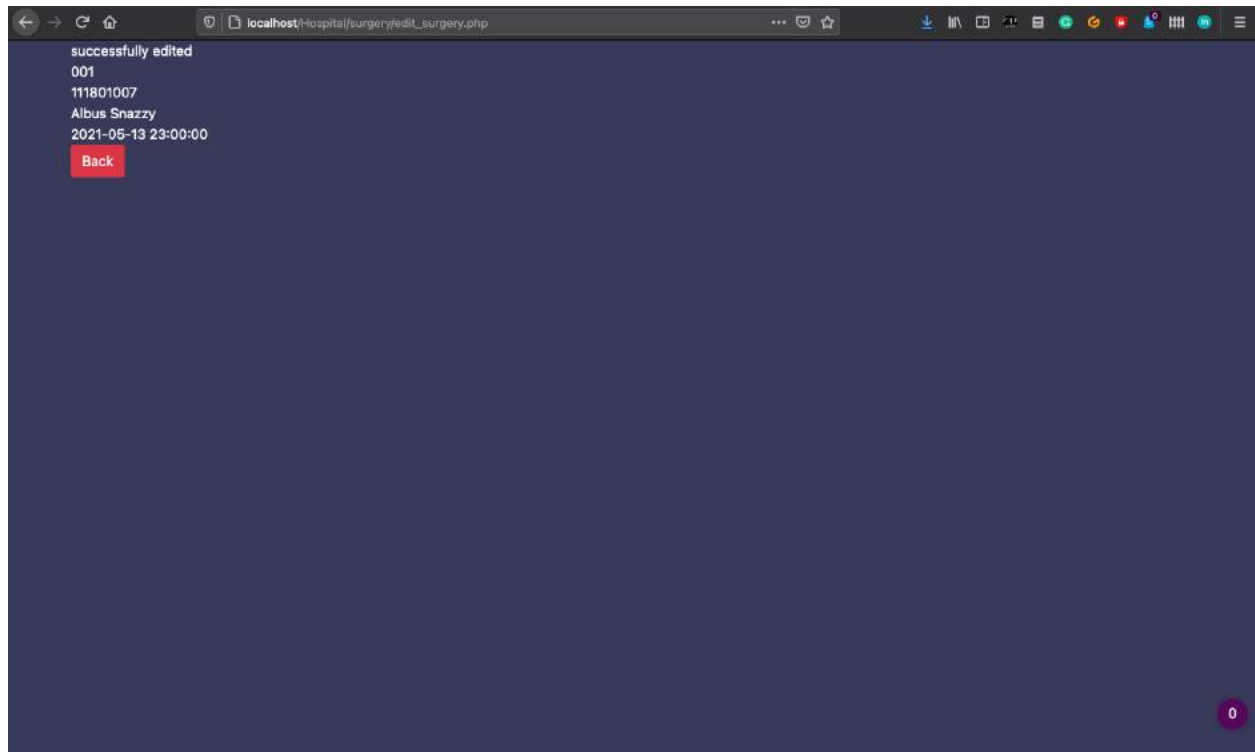
The screenshot shows a web browser window with the address bar displaying 'localhost/hospital/surgery/surgery.php'. The page has a dark blue background and contains a form titled 'Fill out the details to edit'. The form includes four input fields: 'ID' with the value '001', 'Patient ID' with the value '111801007', 'Patient Name' with the value 'Albus Snazzy', and 'Time Of Surgery' with the value '2021-05-13 23:00:00'. A yellow 'submit' button is located below the 'Time Of Surgery' field.

Fill out the details to edit	
ID	001
Patient ID	111801007
Patient Name	Albus Snazzy
Time Of Surgery	2021-05-13 23:00:00

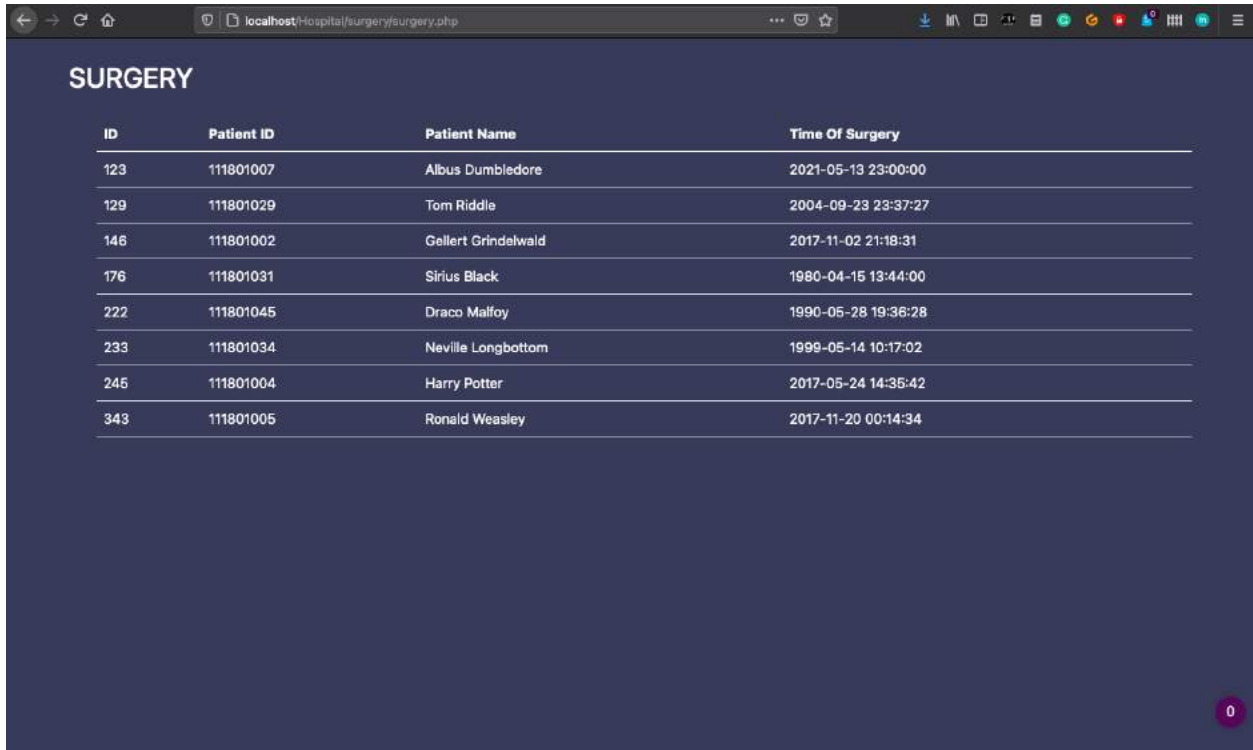
submit

Editing of the preexisting details can also be done, under administrative privileges only.

SURGERY PAGE-EDITING MESSAGE

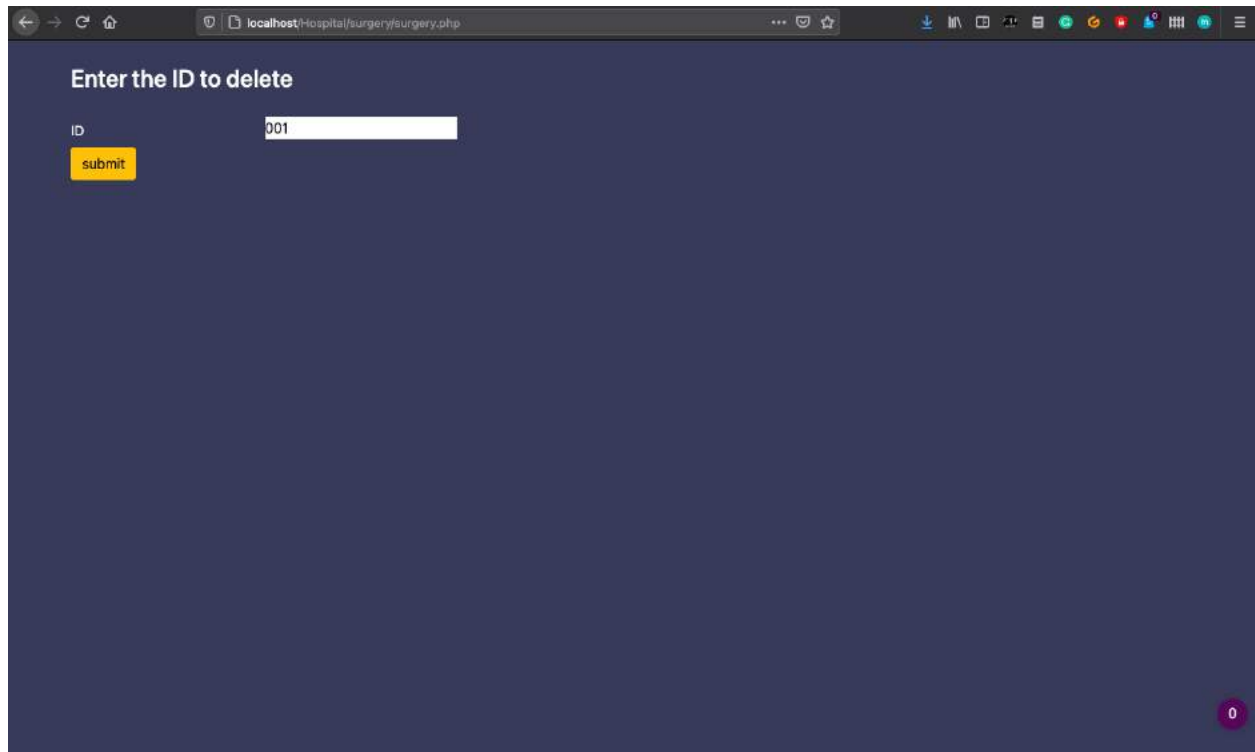


SURGERY PAGE-VIEWING EDITED 'Albus'



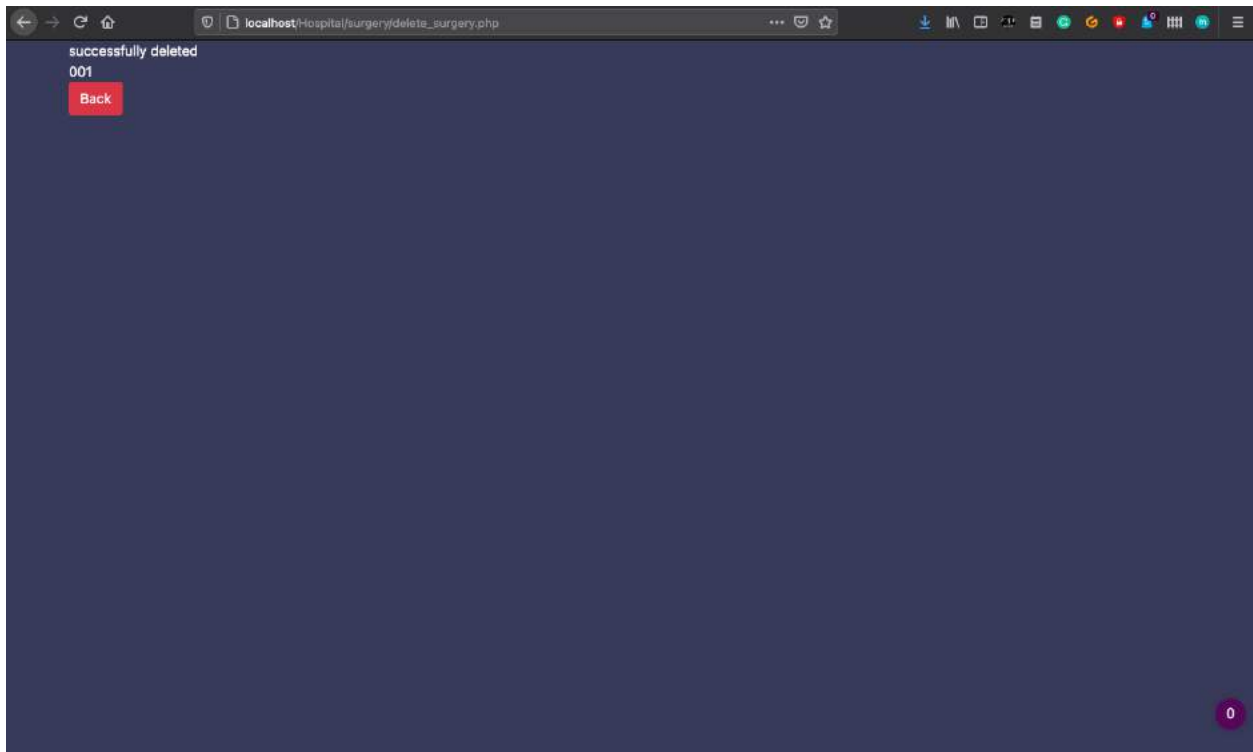
ID	Patient ID	Patient Name	Time Of Surgery
123	111801007	Albus Dumbledore	2021-05-13 23:00:00
129	111801029	Tom Riddle	2004-09-23 23:37:27
146	111801002	Gellert Grindelwald	2017-11-02 21:18:31
176	111801031	Sirius Black	1980-04-15 13:44:00
222	111801045	Draco Malfoy	1990-05-28 19:36:28
233	111801034	Neville Longbottom	1999-05-14 10:17:02
246	111801004	Harry Potter	2017-05-24 14:35:42
343	111801005	Ronald Weasley	2017-11-20 00:14:34

SURGERY PAGE-DELETING 'Albus' ENTRY



A screenshot of a web browser window. The address bar shows 'localhost/hospital/surgery/surgery.php'. The page has a dark blue background. At the top left, the text 'Enter the ID to delete' is displayed. Below this, there is a label 'ID' and a text input field containing '001'. To the left of the input field is a yellow button labeled 'submit'. In the bottom right corner, there is a small purple circular icon with the number '0' inside.

SURGERY PAGE-DELETION MESSAGE



Deletion message will notify the user the status of the deletion operation performed.

CONTRIBUTION TO THE PROJECT

Sai Vipul Mohan V

1. Designed the ERD, schema and all other tasks together.
2. Worked on Surgery and Operation Theatre pages
3. Designed logout functionality in user registration page
4. Designed Home page statistics view

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](#)

Procedures and Views

[Procedures and Views used in the web application](#)