



HOSPITAL MANAGEMENT SYSTEM

CPSC 471: Final Report
Group members: Alokik Singh Kinra(30124487)
Jaskaran Bhatia (30133888)
Shubham Aggarwal(30113383)
Shabnam kaur (30095172)

Abstract

This report outlines the development and implementation of a web application to solve the problem of manual record keeping and lack of online presence for a clinic. The current system is inefficient and error-prone, leading to confusion and lost information for both patients and doctors. The clinic's manual record-keeping system creates challenges for patients when booking appointments and tests. The proposed web application offers a solution for a smoother work-flow system by providing an organized and systematic approach to managing patient records, tracking appointments, and providing information to patients. The application is a user-friendly interface that allows patients to access the clinic's information, book and modify appointments, view the cost of specific operations, and access information on old operations. Doctors can add and edit appointments made by the patients. The proposed system keeps track of all records in databases including costs associated with every operation making the whole process redundant, efficient and less contingent, thus contributing to an increase in the quality of the healthcare provided by the clinic.. The implementation of the web application offers a practical and scalable solution to the problem of manual record keeping and lack of online presence for clinics, ultimately benefiting patients, doctors and administrators alike.

Introduction

The management of employee data can be a daunting task for companies, especially those with a large workforce. Manual record-keeping can lead to errors, misplaced information, and time-consuming data retrieval processes. Furthermore, Patients have to go through the tedious task of booking an appointment through the phone and do not have access to any information about their appointment. In response to this problem, we have designed a database system that centralizes employee data and provides a streamlined approach to managing employee information.

System Description

The database system was developed using HTML, CSS, JavaScript, EJS and MySQL programming languages. The system includes a user-friendly interface that allows authorized users to input and access patient data. Patients can now sign-up easily, book appointments and cancel it too. Administrators can review all the patients and doctors, add operations for a patient and perform changes as required. Doctors can view the list of their appointments and even review patient records to help them with similar medical cases/treatments. This resolves all the aforementioned problems faced by the clinic making it more redundant and time-efficient.

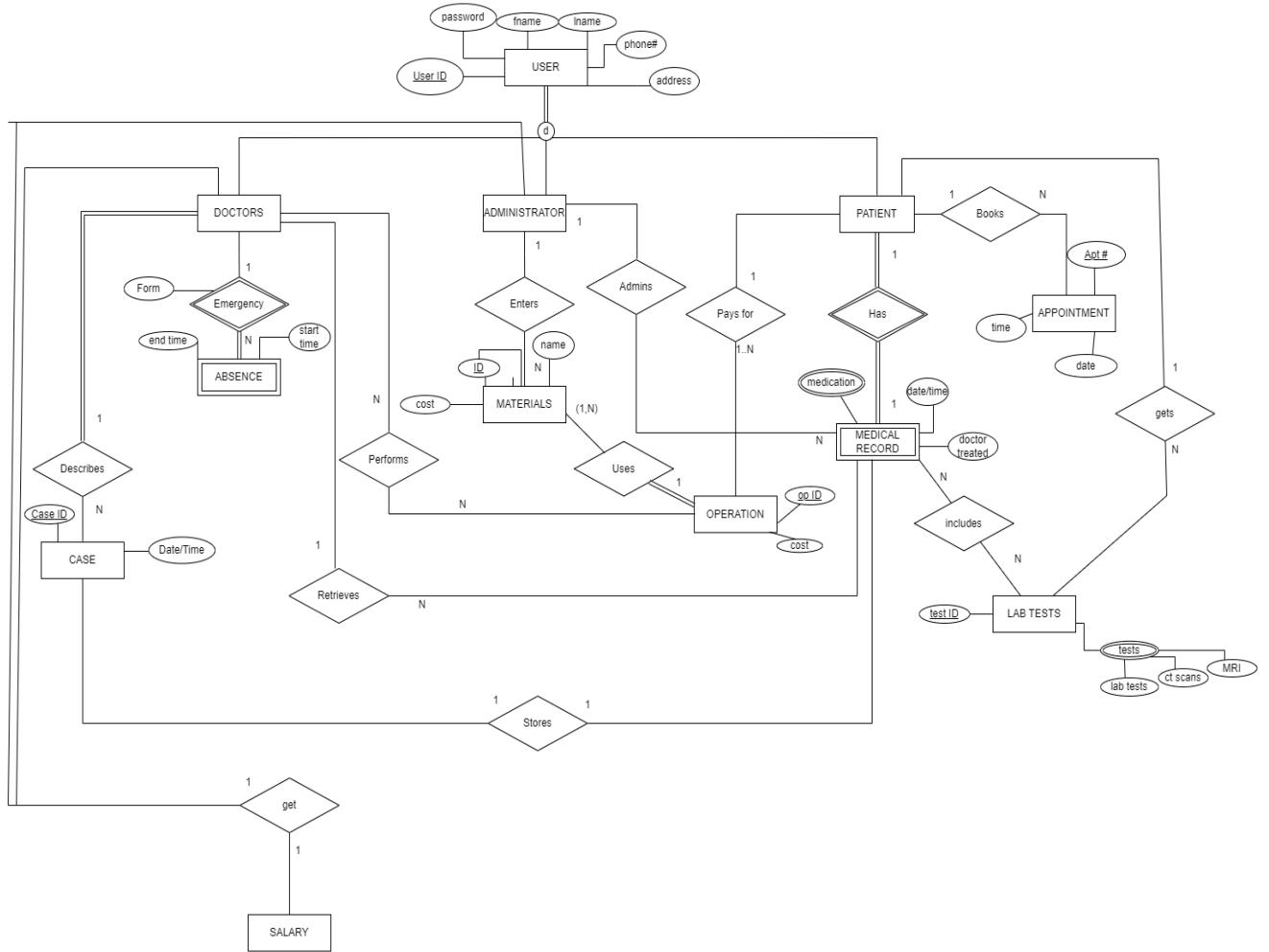
Project Design

The database system is designed to cater to different users with varying access levels. The system has three levels of access: Administrator, Doctor and Employee. The Administrator has full access to the system and can create new users, manage employee data, and generate reports. The Doctor has limited access and can only manage their appointments and also view patient records. The Patient also has limited access allowing them to book/view/cancel appointments with their preferred doctor.

The system is designed to capture all the data from our user-friendly front-end interface and store them in their respective databases. It is assumed that there will only be one head-admin at the very beginning of the application and he can add other existing doctors, administrators and patients. The system also provides functionality to perform data searches and allows the user to filter results on their access level. In conclusion, the database system we have designed provides a centralized platform for managing clinic data allowing for efficient and accurate record-keeping. The system's design allows for varying levels of access to cater to different users, and its functionality to generate appointments assists management in making data-driven decisions.

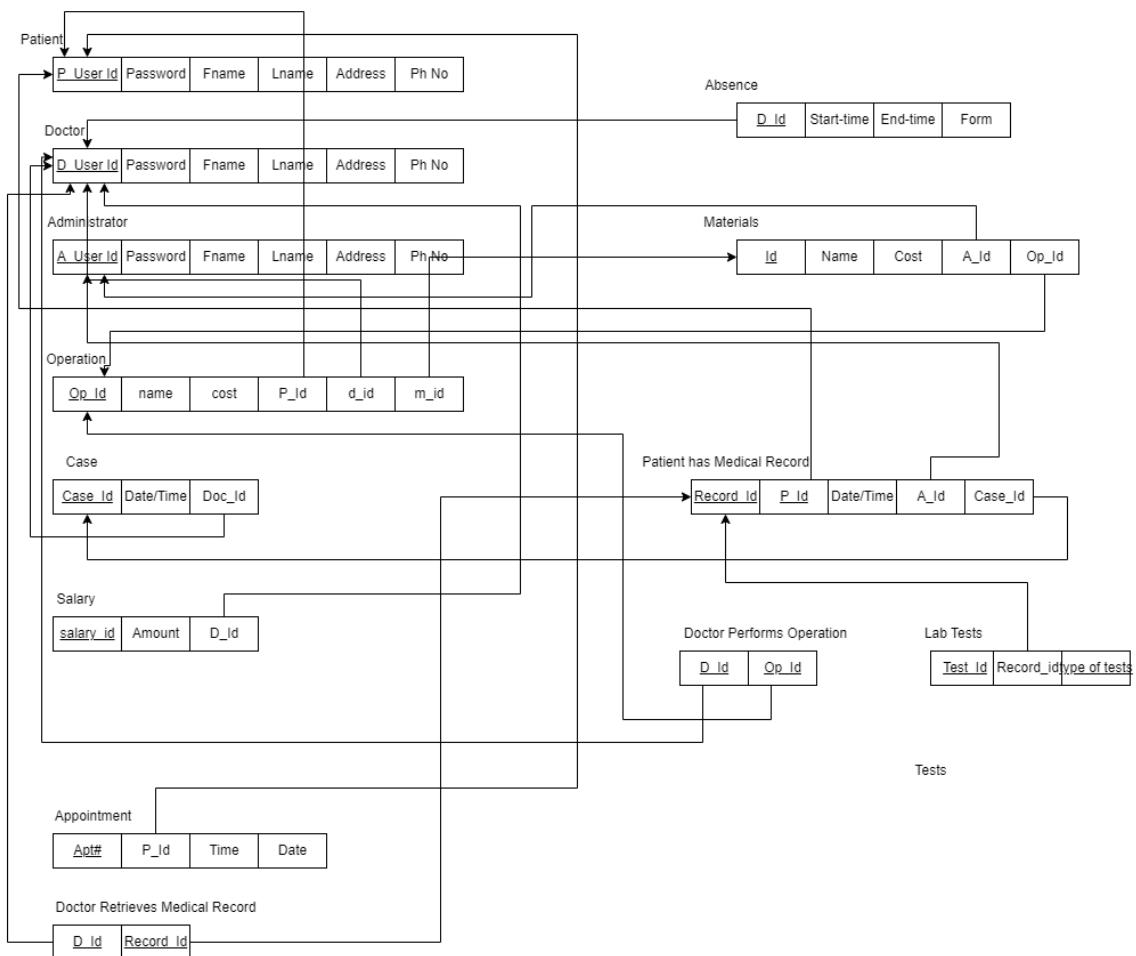
- Patients: Patients are the primary users of the Medical Clinic Management System. They can use the system to book appointments with doctors, and access information about the clinic, doctors, and the services provided. Patients can also cancel or reschedule appointments and view their test results.
- Doctor: Doctors are another crucial user group of the system. They can use the system to view their schedule, manage their appointments, and access the medical history of their patients. Doctors can also prescribe medicines and add medical case information of the patient cured by them.
- Administrators: Administrators play a critical role in the Medical Clinic Management System. They are responsible for managing the clinic's database, overseeing the clinic's operations, and handling patient and doctor records. They can use the system to add or delete doctors and patients, as well as add other administrators. They can manage appointments, add medical operations required for a patient, add Tests(X-ray, CT-scan) and view the financial records.

Updated EERD



So basically our initial EERD has financial records which included salary and bills, but for the updated EERD, we did not include financial records as a whole separate entity but instead we used foreign key references to simplify the scope of the project. Now the salary is a separate entity and it includes salaries of all doctors which have unique ids. Apart from that we also do not have tests now as a separate entity because we included all of those attributes in lab tests and we have another attribute in lab tests which is type of tests.

According to this EERD, we have updated our relational model as well.



QUERIES

```

var mysql = require("mysql");

var express = require("express");

var router = express.Router();

var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "password",
  database: "mydb"
});

router.get("/", function(req, res) {
  var query = "SELECT * FROM Patient";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.post("/addPatient", function(req, res) {
  var query = "INSERT INTO Patient (P_User_Id, Password, Fname, Lname, Address, Ph No) VALUES ('" + req.body.P_User_Id + "','" + req.body.Password + "','" + req.body.Fname + "','" + req.body.Lname + "','" + req.body.Address + "','" + req.body.Ph No + "')";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send("Patient added successfully");
  });
});

router.get("/getPatient/:id", function(req, res) {
  var query = "SELECT * FROM Patient WHERE P_id = " + req.params.id;
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.put("/updatePatient/:id", function(req, res) {
  var query = "UPDATE Patient SET P_User_Id = '" + req.body.P_User_Id + "', Password = '" + req.body.Password + "', Fname = '" + req.body.Fname + "', Lname = '" + req.body.Lname + "', Address = '" + req.body.Address + "', Ph No = '" + req.body.Ph No + "' WHERE P_id = " + req.params.id;
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send("Patient updated successfully");
  });
});

router.delete("/deletePatient/:id", function(req, res) {
  var query = "DELETE FROM Patient WHERE P_id = " + req.params.id;
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send("Patient deleted successfully");
  });
});

router.get("/getDoctors", function(req, res) {
  var query = "SELECT * FROM Doctor";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.get("/getAdministrator", function(req, res) {
  var query = "SELECT * FROM Administrator";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.get("/getOperations", function(req, res) {
  var query = "SELECT * FROM Operation";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.get("/getCases", function(req, res) {
  var query = "SELECT * FROM Case";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.get("/getSalaries", function(req, res) {
  var query = "SELECT * FROM Salary";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.get("/getAppointments", function(req, res) {
  var query = "SELECT * FROM Appointment";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.get("/getAbsences", function(req, res) {
  var query = "SELECT * FROM Absence";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.get("/getMaterials", function(req, res) {
  var query = "SELECT * FROM Materials";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.get("/getRecords", function(req, res) {
  var query = "SELECT * FROM Record";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.get("/getDoctorOperations", function(req, res) {
  var query = "SELECT * FROM Doctor_Performs_Operation";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.get("/getLabTests", function(req, res) {
  var query = "SELECT * FROM Lab_Tests";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.get("/getTests", function(req, res) {
  var query = "SELECT * FROM Tests";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.get("/getPatientRecords", function(req, res) {
  var query = "SELECT * FROM Patient_Has_Medical_Record";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

router.get("/getDoctorRetrievesMedicalRecord", function(req, res) {
  var query = "SELECT * FROM Doctor_Retrieves_Medical_Record";
  con.query(query, function(err, result) {
    if (err) throw err;
    res.send(result);
  });
});

```

```
password: "",  
database: "Hospital_management_system",  
});  
  
con.connect(function (err) {  
  if (err) {  
    throw err;  
  } else {  
    console.log("you are connected");  
  }  
});  
  
module.exports.signup = function (username, email, password, f_name, l_name, Address, Ph_no, callback) {  
  var query =  
    "INSERT INTO `users`(`username`, `email`, `password`, `f_name`, `l_name`, `Address`, `Ph_no`) VALUES (" +  
      username +  
      ", " +  
      email +  
      ", " +  
      password +  
      ", " +  
      f_name +  
      ", " +  
      l_name +
```



```
"", "+  
ph_no +  
");  
con.query(query, callback);  
};  
  
module.exports.signup_patient = function (username, password, f_name, l_name, email, address, ph_no, callback) {  
var query =  
"INSERT INTO `patient`(`username`, `password`, `f_name`, `l_name`, `email`, `address`, `ph_no`) VALUES ("  
+  
username +  
", "+  
password +  
", "+  
f_name +  
", "+  
l_name +  
", "+  
email +  
", "+  
address +  
", "+  
ph_no +
```



```
};
```

```
module.exports.add_doctor = function (username, password, first_name, last_name, email, address, phone, callback) {  
    var query = "INSERT INTO `doctor`(`username`, `password`, `first_name`, `last_name`, `email`, `address`, `phone`) values (" +  
        username +  
        ", " +  
        password +  
        ", " +  
        first_name +
```

```
"", "+  
last_name +  
", "+  
email +  
", "+  
address +  
", "+  
phone +  
");  
console.log(query);  
con.query(query, callback);  
};  
  
module.exports.add_case = function (  
  
date,  
time,  
d_id,  
info,  
callback  
{  
var query =  
"INSERT INTO `Casedoctor`(`date`,`time`,`d_id`, `info`) values ('" +  
date +
```

```
"", "+  
time +  
", "+  
d_id +  
", "+  
info +  
");  
console.log(query);  
con.query(query, callback);  
};  
  
module.exports.add_patients = function (  
    username,  
    password,  
    f_name,  
    l_name,  
    email,  
    address,  
    ph_no,  
    callback  
) {  
    var query =  
        "INSERT INTO `patient`(`username`, `password`, `f_name`, `l_name`, `email`, `address`, `ph_no`) values ('" +
```

```
username +  
", " +  
password +  
", " +  
f_name +  
", " +  
l_name +  
", " +  
email +  
", " +  
address +  
", " +  
ph_no +  
");  
console.log(query);  
con.query(query, callback);  
};
```

```
module.exports.add_record = function (  
    p_id,  
    date_time,  
    a_id,  
    case_id,  
    callback
```

```
){

var query =
    "INSERT INTO `patient has medical record`(`p_id`, `date_time`, `a_id`, `case_id`) values (" +
    p_id +
    ", " +
    date_time +
    ", " +
    a_id +
    ", " +
    case_id +
    ")";
console.log(query);
con.query(query, callback);
};
```

```
module.exports.add_test = function (
    record_id,
    type_of_tests,
    callback
) {
    var query =
        "INSERT INTO `lab tests`(`record_id`, `type_of_tests`) values (" +
        record_id +
        ", " +
```

```
type_of_tests +  
");  
  
console.log(query);  
  
con.query(query, callback);  
};  
  
  
module.exports.getAllpatients = function (callback) {  
  
var query = "select * from patient";  
  
con.query(query, callback);  
};  
  
  
module.exports.getAllRecords = function (callback) {  
  
var query = "select * from `patient has medical record`";  
  
con.query(query, callback);  
};  
  
  
module.exports.getAllTests = function (callback) {  
  
var query = "select * from `lab tests`";  
  
con.query(query, callback);  
};  
  
  
module.exports.getpatientsbyId = function (P_userid, callback) {  
  
var query = "select * from patient where P_userid =" + P_userid;
```

```
con.query(query, callback);

};

module.exports.getRecordsbyId = function (record_id, callback) {

var query = "select * from `patient has medical record` where record_id =" + record_id;

con.query(query, callback);

};

module.exports.getTestsbyId = function (test_id, callback) {

var query = "select * from `lab tests` where test_id =" + test_id;

con.query(query, callback);

};

module.exports.getpatient = function (P_userid, callback) {

var query = "select P_userid from patient where P_userid =" + P_userid;

con.query(query, callback);

};

module.exports.getRecord = function (record_id, callback) {

var query = "select record_id from `patient has medical record` where record_id =" + record_id;

con.query(query, callback);

};
```

```
module.exports.add_materials = function(  
    name,  
    cost,  
    a_id,  
    op_id,  
    callback  
) {  
  
    var query =  
  
        "INSERT INTO `materials`(`name`, `cost`, `a_id`, `op_id`) values ('" +  
        name +  
        "', '" +  
        cost +  
        "', '" +  
        a_id +  
        "', '" +  
        op_id +  
        "')";  
  
    console.log(query);  
  
    con.query(query, callback);  
};
```

```
module.exports.add_operation = function(  
    //op_id,  
    op_name,
```

```
cost,  
p_id,  
d_id,  
materials_id,  
callback  
{  
  
var query =  
  
"INSERT INTO `operation`(`op_name`,`cost`,`p_id`,`d_id`, `materials_id`) values (" +  
// op_id +  
// "", "" +  
op_name +  
", "", "+  
cost +  
", "", "+  
p_id +  
", "", "+  
d_id +  
", "", "+  
materials_id +  
")";  
  
console.log(query);  
  
con.query(query, callback);  
};
```

```
module.exports.getAlloperation = function (callback) {  
    var query = "select * from operation";  
    con.query(query, callback);  
};  
  
module.exports.getoperationbyId = function (op_id, callback) {  
    var query = "select * from operation where op_id =" + op_id;  
    con.query(query, callback);  
};  
  
module.exports.edit_operation = function (  
    op_id,  
    op_name,  
    cost,  
    p_id,  
    d_id,  
    materials_id,  
    callback  
) {  
    var query =  
        "update `operation` set `op_name`=''" +  
        op_name +  
        "'", `cost`=''" +  
        cost +
```

```
"', `p_id`=" +  
p_id +  
"', `d_id`=" +  
d_id +  
"', `materials_id`=" +  
materials_id +  
"' where op_id=" +  
op_id;  
  
con.query(query, callback);  
  
// console.log(query);  
  
};  
  
  
module.exports.delete_operation = function (op_id, callback) {  
  
var query = "delete from operation where op_id=" + op_id;  
  
con.query(query, callback);  
  
};  
  
  
module.exports.delete_salary = function (salary_id, callback) {  
  
var query = "delete from salary where salary_id=" + salary_id;  
  
con.query(query, callback);  
  
};  
  
  
module.exports.getAllmaterials = function (callback) {  
  
var query = "select * from materials";
```

```
con.query(query, callback);

};

module.exports.getMaterialsbyId = function (id, callback) {

var query = "select * from materials where id =" + id;

con.query(query, callback);

};

module.exports.getAllAdministrator = function (callback) {

var query = "select * from administrator";

con.query(query, callback);

};

module.exports.getAllDoc = function (callback) {

var query = "select * from doctor";

con.query(query, callback);

};

module.exports.getAllCase = function (callback) {

var query = "select * from Casedoctor";

con.query(query, callback);

};

module.exports.getDocbyId = function (d_id, callback) {

var query = "select * from doctor where d_id =" + d_id;

con.query(query, callback);

};
```

```
};
```

```
module.exports.getCasebyId = function (case_id, callback) {  
  
    var query = "select * from Casedoctor where case_id =" + case_id;  
  
    con.query(query, callback);  
  
};
```

```
module.exports.getAdministratorbyId = function (a_id, callback) {  
  
    var query = "select * from administrator where a_id =" + a_id;  
  
    con.query(query, callback);  
  
};
```

```
module.exports.editDoc = function (  
    d_id,  
    username,  
    password,  
    first_name,  
    last_name,  
    email,  
    address,  
    phone,  
    callback
```

```
) {  
  
    var query =  
  
        "update `doctor` set `username`=''" +  
  
        username +  
  
        ",'`password`=''" +  
  
        password +  
  
        ",'`first_name`=''" +  
  
        first_name +  
  
        ",'`last_name`=''" +  
  
        last_name +  
  
        ",'`email`=''" +  
  
        email +  
  
        ",'`address`=''" +  
  
        address +  
  
        ",'`phone`=''" +  
  
        phone +  
  
        "' where d_id=" +  
  
        d_id;  
  
    con.query(query, callback);  
  
    // console.log(query);  
  
};
```

```
module.exports.editCase = function (case_id, date, time, d_id, info, callback) {
    var query =
        "update `Casedoctor` set `date`=" + date +
        ", `time`=" + time +
        ", `d_id`=" + d_id +
        ", `info`=" + info +
        " where case_id=" +
        case_id;
    con.query(query, callback);
    // console.log(query);
};
```

```
module.exports.edit_patients = function (P_userid,
username,
password,
f_name,
l_name,
email,
address,
ph_no,
callback
){

var query =
"update `patient` set `username`='"
+
username +
"', `password`='"
+
password +
"', `f_name`='"
+
f_name +
"', `l_name`='"
+
l_name +
"', `email`='"
+
email +
"', `address`='"
+
address +
```

```
"`ph_no`="" +
ph_no +
"" where P_userid=" +
P_userid;
con.query(query, callback);

};
```

```
module.exports.edit_record = function (
record_id,
p_id,
date_time,
a_id,
case_id,
callback
) {
var query =
"update `patient has medical record` set `p_id`="" +
p_id +
"" , `date_time`="" +
date_time +
"" , `a_id`="" +
a_id +
"" , `case_id`="" +
```

```
        case_id +  
        "' where record_id=" +  
        record_id;  
  
    con.query(query, callback);  
  
};  
  
module.exports.edit_test = function (  
    test_id,  
    record_id,  
    type_of_tests,  
    callback  
) {  
  
    var query =  
        "update `lab tests` set `record_id`='"+  
        record_id +  
        "', `type_of_tests`='"+  
        type_of_tests +  
        "' where test_id=" +  
        test_id;  
  
    con.query(query, callback);  
  
};
```

```
module.exports.edit_materials = function ( id, name, cost, a_id, op_id, callback ) {  
  
    var query =  
        "update `materials` set `name`='\" +  
        name +  
        "\", `cost`='\" +  
        cost +  
        "\", `a_id`='\" +  
        a_id +  
        "\", `op_id`='\" +  
        op_id +  
        "\" where id=\" +  
        id;  
  
    con.query(query, callback);  
  
};
```

```
module.exports.editAdministrator = function (a_id, username, email, password, f_name, l_name, address, ph_no, callback) {  
    var query = "update `administrator` set `username`=''" +  
        username +  
        "', `email`=''" +  
        email +  
        "', `password`=''" +  
        password +  
        "', `f_name`=''" +  
        f_name +  
        "', `l_name`=''" +  
        l_name +  
        "', `address`=''" +  
        address +
```

```
"',ph_no`="" +  
ph_no +  
" where a_id=" +  
a_id;  
con.query(query, callback);  
  
};  
  
module.exports.deleteDoc = function (d_id, callback) {  
  
var query = "delete from doctor where d_id=" + d_id;  
con.query(query, callback);  
};  
  
module.exports.deleteCase = function (case_id, callback) {  
  
var query = "delete from Casedoctor where case_id=" + case_id;  
con.query(query, callback);  
};  
  
module.exports.delete_patients = function (P_userid, callback) {
```

```
var query = "delete from patient where P_userid=" + P_userid;
con.query(query, callback);
};

module.exports.delete_record = function (record_id, callback) {

var query = "delete from `patient has medical record` where record_id=" + record_id;
con.query(query, callback);
};

module.exports.delete_test = function (test_id, callback) {

var query = "delete from `lab tests` where test_id=" + test_id;
con.query(query, callback);
};

module.exports.deleteAdministrator = function (a_id, callback) {

var query = "delete from administrator where a_id=" + a_id;
con.query(query, callback);
};

module.exports.deleteMaterials = function (id, callback) {
```

```
var query = "delete from materials where id=" + id;  
con.query(query, callback);  
};  
  
module.exports.add_appointment = function (  
    p_id,  
    p_name,  
    d_name,  
    date,  
    time,  
    callback  
) {  
  
    var query =  
        "insert into appointment (p_id,p_name,d_name,date,time) values ('" +  
        p_id +  
        "','" +  
        p_name +  
        "','" +  
        d_name +  
        "','" +  
        date +
```

```
    ", " +  
  
    time +  
  
    ");  
  
con.query(query, callback);  
  
};
```

```
module.exports.getallappointment = function (callback) {  
    var query = "select * from appointment";  
  
    con.query(query, callback);  
  
};
```

```
module.exports.getPatientappointment = function ( callback ) {  
  
    var query = "select * from appointment, patient where patient.P_userid = appointment.p_id";  
  
    con.query(query, callback);  
  
};
```

```
module.exports.searchDoc = function (key, callback) {  
  
    var query = 'SELECT *from doctor where first_name like "%' + key + '%"';  
  
    con.query(query, callback);  
  
    console.log(query);  
  
};
```

```
module.exports.searchCase = function (key, callback) {  
  
    var query = 'SELECT *from Casedoctor where d_id like "%" + key + "%"';  
  
    con.query(query, callback);  
  
    console.log(query);  
  
};
```

```
module.exports.search_operation = function (op_name, callback) {  
  
    var query = 'SELECT *from operation where op_name like "%" + op_name + "%"';  
  
    con.query(query, callback);  
  
    console.log(query);  
  
};
```

```
module.exports.searchoperations = function (d_id, callback) {  
  
    var query = 'SELECT *from operation where d_id like "%" + d_id + "%"';  
  
    con.query(query, callback);  
  
    console.log(query);  
  
};
```

```
module.exports.searchPatients = function (f_name, callback) {  
  var query = 'SELECT *from patient where f_name like "%" + f_name + "%"';  
  con.query(query, callback);  
  console.log(query);  
};  
  
module.exports.searchRecord = function (record_id, callback) {  
  var query = 'SELECT *from `patient has medical record` where record_id like "%" + record_id + "%"';  
  con.query(query, callback);  
  console.log(query);  
};  
  
module.exports.searchTest = function (test_id, callback) {  
  var query = 'SELECT *from `lab tests` where test_id like "%" + test_id + "%"';  
  con.query(query, callback);  
  console.log(query);  
};  
  
module.exports.searchmaterials = function (name, callback) {  
  var query = 'SELECT *from materials where name like "%" + name + "%"';  
  con.query(query, callback);  
};
```

```
module.exports.searchAdministrator = function (f_name, callback) {  
    var query = 'SELECT *from administrator where f_name like "%" + f_name + "%"';  
    con.query(query, callback);  
    // console.log(query);  
};  
  
module.exports.searchAppointment = function (p_id, callback) {  
    var query = "SELECT *from appointment where p_id = " + p_id ;  
    con.query(query, callback);  
    // console.log(query);  
};  
  
module.exports.searchAppointment_bydoc = function (d_name, callback) {  
    var query = "SELECT *from appointment where d_name = " + d_name ;  
    con.query(query, callback);  
    // console.log(query);  
};  
  
module.exports.searchAbsence = function (d_id, callback) {  
    var query = 'SELECT *from absence where d_id like "%" + d_id + "%"';  
    con.query(query, callback);  
    // console.log(query);
```

```
};
```

```
module.exports.getappointmentbyid = function (a_id, callback) {  
  
    var query = "select * from appointment where a_id=" + a_id;  
  
    console.log(query);  
  
    con.query(query, callback);  
  
};
```

```
module.exports.getappointmentbypatientid = function (p_id, callback) {  
  
    var query = "select * from appointment where p_id=" + p_id;  
  
    console.log(query);  
  
};
```

```
module.exports.edit_appointment = function (  
    a_id,  
    p_id,  
    p_name,  
    d_name,  
    date,  
    time,  
    callback  
) {  
  
    var query =  
        "update appointment set p_id=''" +
```

```
p_id +  
",p_name=" +  
p_name +  
",d_name=" +  
d_name +  
",date=" +  
date +  
",time=" +  
time +  
" where a_id=" +  
a_id;  
con.query(query, callback);  
};  
  
module.exports.delete_appointment = function (a_id, callback) {  
var query = "delete from appointment where a_id=" + a_id;  
con.query(query, callback);  
};  
//module.exports =router;  
  
module.exports.setpassword = function (id, newpassword, callback) {
```

```
var query =  
  "update `users` set `password`=''' + newPassword + ''' where id=" + id;  
  
con.query(query, callback);  
};
```

```
module.exports.add_administrator = function (
```

```
  username,
```

```
  email,
```

```
  password,
```

```
  f_name,
```

```
  l_name,
```

```
  address,
```

```
  ph_no,
```

```
  callback
```

```
) {
```

```
  var query =
```

```
    "Insert into `administrator`(`username`, `email`, `password`, `f_name`, `l_name`, `address`, `ph_no`) values ('"  
  +
```

```
  username +
```

```
  "', '' +
```

```
  email +
```

```
  "', '' +
```

```
  password +
```

```
  "', '' +
```

```
  f_name +
```

```
"", "+  
l_name +  
", "+  
address +  
", "+  
ph_no +  
");  
con.query(query, callback);  
console.log(query);  
};  
  
module.exports.addMed = function (  
name,  
p_date,  
expire,  
e_date,  
price,  
quantity,  
callback  
) {  
var query =  
"Insert into `store` (name,p_date,expire,expire_end,price,quantity) values('" +  
name +  
", "+
```

```
p_date +  
", " +  
expire +  
", " +  
e_date +  
", " +  
price +  
", " +  
quantity +  
");  
console.log(query);  
con.query(query, callback);  
};  
  
module.exports.getMedbyId = function (id, callback) {  
var query = "select * from store where id=" + id;  
con.query(query, callback);  
};  
  
module.exports.editmed = function (  
id,  
name,  
p_date,  
expire,
```

```
e_date,  
price,  
quantity,  
callback  
) {  
  
var query =  
  
"update store set name=' " +  
name +  
" ', p_date=' " +  
p_date +  
" ',expire=' " +  
expire +  
" ',expire_end=' " +  
e_date +  
" ',price=' " +  
price +  
" ',quantity=' " +  
quantity +  
" ' where id=" +  
id;  
  
console.log(query);  
con.query(query, callback);  
};
```

```
module.exports.getadministrator = function (callback) {  
    var query = "select * from administrator";  
    con.query(query, callback);  
};  
  
module.exports.add_absence = function (  
    d_id,  
    starttime,  
    endtime,  
    form,  
    callback  
) {  
    var query =  
        "Insert into `absence` (`d_id`,`starttime`,`endtime`,`form`) values ('" +  
        d_id +  
        "','" +  
        starttime +  
        "','" +  
        endtime +  
        "','" +  
        form +  
        "')";
```

```
console.log(query);

con.query(query, callback);

};

module.exports.getAllabsence = function (callback) {

var query = "Select * from absence";

con.query(query, callback);

};

module.exports.getpatientdetails = function (username, callback) {

var query = "select * from patient where username='" + username + "'";

con.query(query, callback);

console.log(query);

};

module.exports.getadmindetails = function (username, callback) {

var query = "select * from administrator where username='" + username + "'";

con.query(query, callback);

console.log(query);

};
```

```
module.exports.getdoctordetails = function (username, callback) {  
  
    var query = "select * from doctor where username='" + username + "'";  
  
    con.query(query, callback);  
  
    console.log(query);  
  
};
```

```
module.exports.getcasedetails = function (case_id, callback) {  
  
    var query = "select * from Casedoctor where case_id='" + case_id + "'";  
  
    con.query(query, callback);  
  
    console.log(query);  
  
};
```

```
module.exports.patient_profile = function (  
    P_userid,  
    username,  
    email,  
    password,  
    callback  
) {
```

```
    var query =  
        "update patient set username ='" +  
        username +
```

```
        "", email = "" +  
        email +  
        "",password="" +  
        password +  
        "" where P_userid=" +  
        P_userid;  
        con.query(query, callback);  
        console.log(query);  
    };
```

```
module.exports.edit_profile = function (  
    a_id,  
    username,  
    email,  
    password,  
    callback  
) {  
    var query =  
        "update administrator set username ='" +  
        username +  
        "', email ='" +  
        email +  
        "',password=' +
```



```
con.query(query, callback);

console.log(query);

};

module.exports.edit_patprofile = function (
  P_userid,
  username,
  email,
  password,
  callback
) {
  var query =
    "update patient set username ='" +
    username +
    "', email ='" +
    email +
    "',password=''" +
    password +
    "' where P_userid=" +
    P_userid;

  con.query(query, callback);

  console.log(query);

};
```

```
module.exports.getabsencebyid = function (leave_id, callback) {  
    var query = "select * from absence where leave_id=" + leave_id;  
    con.query(query, callback);  
};  
  
module.exports.delete_absence= function (leave_id, callback) {  
    var query = "delete from absence where leave_id=" + leave_id;  
    con.query(query, callback);  
};  
  
module.exports.edit_absence = function (  
    leave_id,  
    d_id,  
    starttime,  
    endtime,  
    form,  
    callback  
) {  
    var query =  
        "update absence set d_id='\" +
```

```
d_id +  
",starttime=" +  
starttime +  
",endtime=" +  
endtime +  
",form=" +  
form +  
" where leave_id=" +  
leave_id;  
con.query(query, callback);  
};
```

```
module.exports.add_salary = function (  
    amount,  
    d_id,  
    callback  
) {  
  
    var query =  
        "INSERT INTO `salary` (amount, d_id) VALUES (" +  
        amount +  
        ", " +  
        d_id +  
        ");  
  
    console.log(query);
```

```
con.query(query, callback);

};

module.exports.edit_salary = function (
    salary_id,
    amount,
    d_id,
    callback
) {
    var query =
        "update `salary` set `amount`=" +
        amount +
        ", `d_id`=" +
        d_id +
        " where salary_id=" +
        salary_id;

    con.query(query, callback);
    // console.log(query);
};

module.exports.getAllsalary = function (callback) {
    var query = "select * from salary";
    con.query(query, callback);
};
```

```
module.exports.getsalarybyId = function (salary_id, callback) {  
    var query = "select * from salary where salary_id =" + salary_id;  
    con.query(query, callback);  
};  
  
module.exports.search_salary = function (key, callback) {  
    var query = 'SELECT *from salary where d_id like "%' + key + '%"';  
    con.query(query, callback);  
    console.log(query);  
};
```

Description of the visual interface highlighting its friendliness and usability

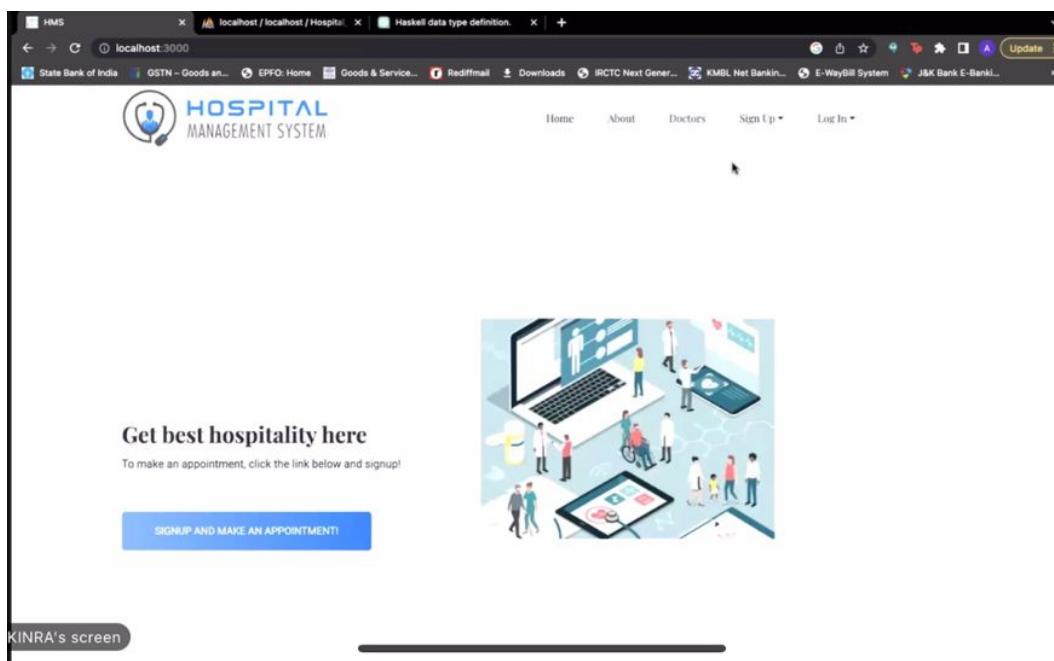


Fig1: Home Screen

A screenshot of a web browser window titled "localhost:3000/#abt". The main content area features a large blue circular graphic at the top. Below it, the heading "About us" is displayed in bold black font. A descriptive paragraph follows: "We are creating a hospital management system that will provide the best services to patients, doctors and administrators and will be able to manage the hospital in a better way."

Our top-ranked doctors



A screenshot of a web browser window titled "localhost:3000/#doc". The content is identical to the "About us" screen, featuring four placeholder icons for doctor profiles and their names listed below them.

Our top-ranked doctors



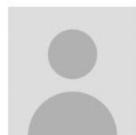
Aksh



Sukhansh



Sukhdeep

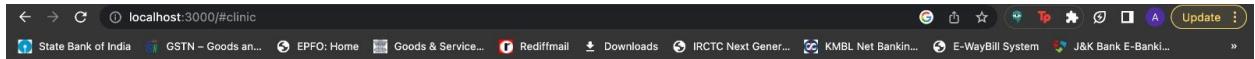


Aron

Heer

Shubh

Muhammad



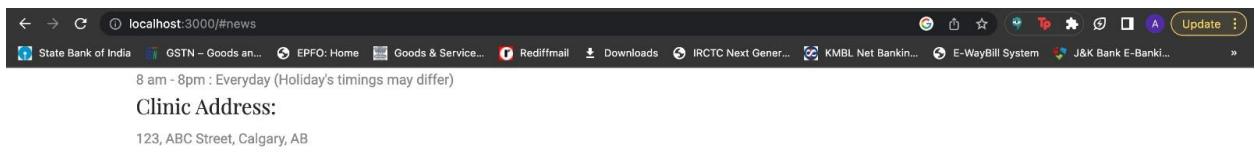
Clinic Operation Timings:

8 am - 8pm : Everyday (Holiday's timings may differ)

Clinic Address:

123, ABC Street, Calgary, AB

A screenshot of a newsletter sign-up form. It features a placeholder text "Your Email Address" in a white input field, a red "X" button to its right, and a large red "Submit" button below it. The background is light gray.



Newsletter

Stay update with our latest

Enter your email address to join our newsletter!

Your Email Address



Copyright © 2023 All rights reserved | This template is referenced from Colorlib

The visual interface for this project is designed to be user-friendly and easy to navigate. The interface is clean and simple, with a modern and intuitive design that makes it easy for users to quickly find what they're looking for. The navigation is easy to understand and consistent throughout the interface.

The login and sign up buttons are clearly visible at the top right hand side of the Home page which has a drop down menu where you can select whether you are a doctor, patient or an administrator. There is also a sign-up and make an appointment option button on the home page which is easily accessible for new patients. There is also an about us and doctors page for patients to discover. The labels and intuitive icons are easy to understand. The interface is clean and uncluttered which avoids unnecessary distractions. The interface is designed to prevent errors.

The interface is organized in a logical and intuitive way, with a clear hierarchy of information that helps users understand how to use the tool.

The interface is designed with usability in mind, with clear labeling and easy-to-understand icons that make it easy for users to quickly understand what each function does. The layout is consistent throughout the interface and matches user expectations, which helps users feel comfortable and confident as they navigate the tool.

The font and color scheme utilized in the visual interface are significant components of its usability. The chosen font type is designed to optimize legibility, while the color scheme has been carefully selected to not only enhance the aesthetic appeal of the interface, but also ensure readability and provide appropriate contrast. These elements have been thoughtfully incorporated to align with the concept of the project and enhance its overall functionality.

Quick User Guide

For Patients:

Starting with the home screen(Fig1) the patient has a variety of options available to obtain information about the clinic and doctors. If the patient does not have an account ,they have to first sign up with their Contact information in order to obtain their credentials by clicking sign up and **make an appointment button** on the homescreen.(Fig2).If the patient already has an existing account, they can navigate directly to the Login Page by clicking Login at the bottom of the Sign-Up Page(Fig 2).Next, they can enter their login username and password in the login page to successfully enter their account portal.(Fig 3).

localhost:3000/signup_patient

of India GSTN – Goods an... EPFO: Home Goods & Service... Redifmail

Dimensions: iPhone XR ▾ 414 x 896 78% ▾ No throttling

Username

Password

First name

Last Name

email

Address

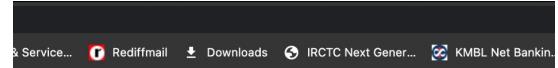
Phone Number

I have read and agree the Terms & Conditions

SIGNUP

Already have an account? [Login](#)
[Forgot Password](#)

Sign up page screenshot(patient)(Fig2)



Patient Login Form

username

Password

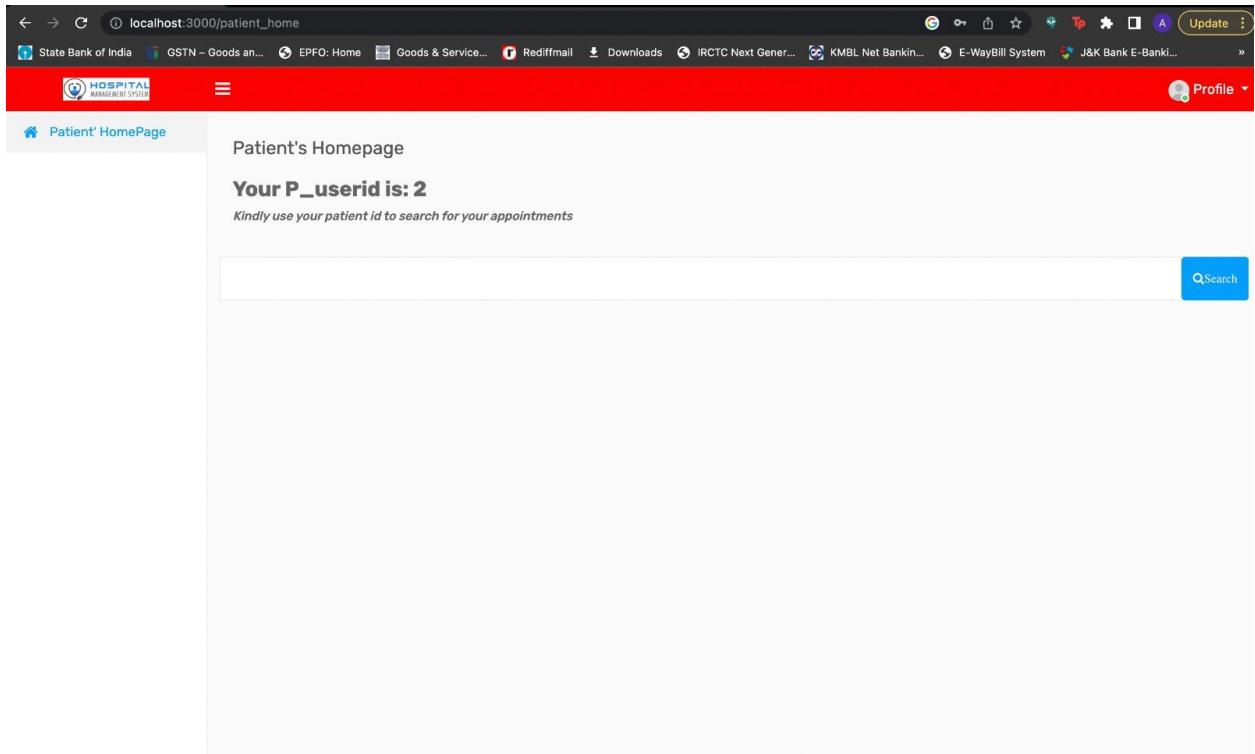
Log In

[Sign Up](#)

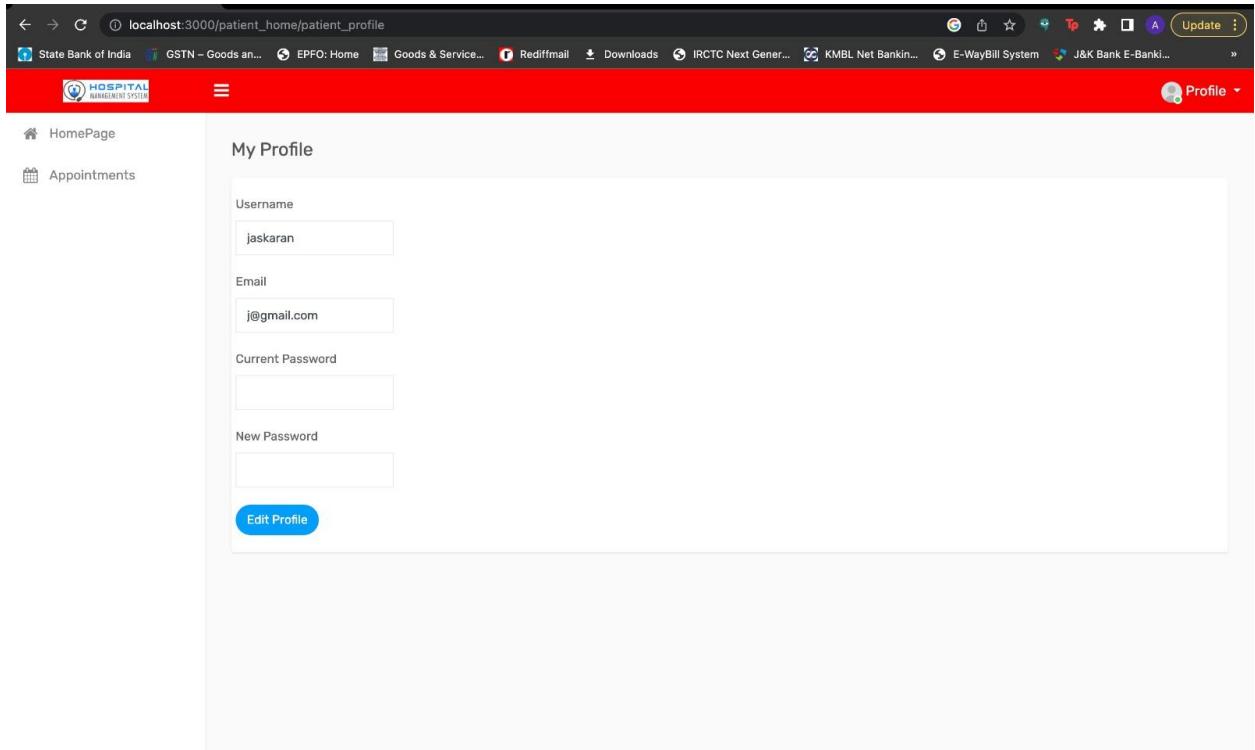
[Forgot Password](#)

Login page Patient(Fig3)

Once logged in, the patient will have the following homepage that displays the patient id.



The profile icon in the top right corner lets the patient access a from down menu from where they can either logout or edit profile.



The user is prompted to put in their patient id in order to access their appointments. If they don't have any appointments, they can make a new one on the next screen.

localhost:3000/patient_appointment/p_add_appointment/

HOSPITAL MANAGEMENT SYSTEM

Patient's Appointments

Add Appointment

| | |
|-------------|-----------------------------------|
| Patient ID | Patient Name |
| 2 | Jaskaran |
| Doctor Name | Date |
| aama | 04/04/2023 |
| Time | |
| 6:22 PM | (<input type="button" value=""/> |

CREATE APPOINTMENT

localhost:3000/patient_home/search

HOSPITAL MANAGEMENT SYSTEM

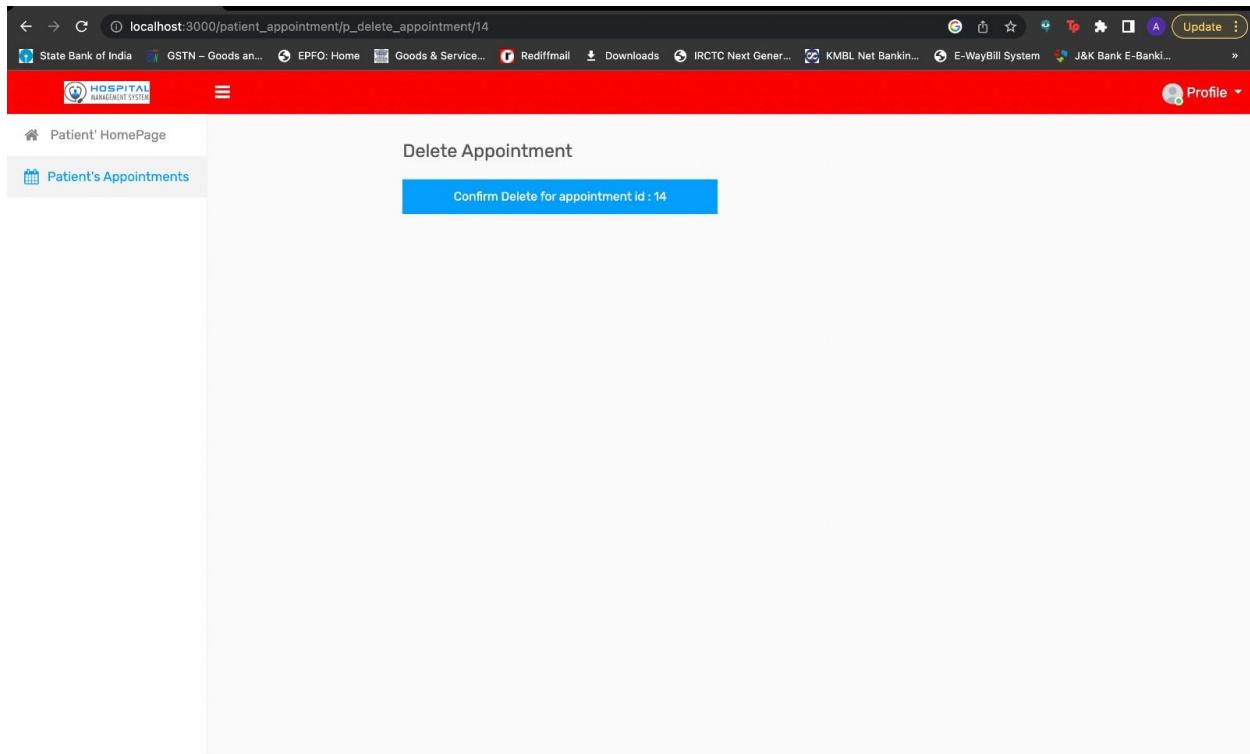
Patient's Appointments

Appointments

+ Add Appointment

| Appointment ID | Patient ID | Patient Name | Doctor name | Appointment date | Appointment time | More |
|----------------|------------|---------------|-------------|------------------|------------------|---------------------------------|
| 14 | 2 | aqsw | sukhdeep | 11/04/2023 | 1:30 PM | <input type="button" value=""/> |
| 20 | 2 | AS | sukhdeep | 28/03/2023 | 4:04 PM | <input type="button" value=""/> |
| 23 | 2 | jaskaransingh | love | 12/04/2023 | 5:38 AM | <input type="button" value=""/> |
| 30 | 2 | Jaskaran | aama | 04/04/2023 | 6:22 PM | <input type="button" value=""/> |

Patient can also edit/delete appointments from the menu under "more."



For Doctors:

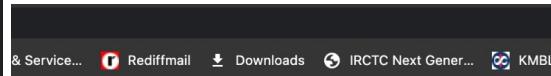
Most of the doctors are added by the admin already, but for new Doctors who do not have an existing account, the Sign-Up Process is very similar to the patient's sign-up.

Next, they can login to their portal through the Doctors Login Page as shown below.

The screenshot shows a mobile browser window with the URL `localhost:3000/signup_doctor`. The page title is "Doctor Sign Up". The form fields include:

- Username (input field)
- Password (input field)
- First Name (input field)
- Last Name (input field)
- Email (input field)
- Address (input field)
- Phone Number (input field)
- A checkbox labeled "I have read and agree the Terms & Conditions".
- A blue "SIGNUP" button.
- Links at the bottom: "Already have an account? Login" and "Forgot Password".

Doctor's Sign UP Page



Doctor's Login Page

Next, They will be directed to their homepage with their ID. The menu on the left lets them access their appointments, Doctors salary, medical records, cases and operations. Doctor's can use the profile icon on the top right corner to access a drop down menu from where they can either logout or go to edit profile.

localhost:3000/doctor_home

HOSPITAL MANAGEMENT SYSTEM

Profile

Doctor's HomePage

Your Doctor's ID is: 228

Appointments

Doctor's Salary

Medical Records

Cases

Operations to perform

Doctors have access to the appointments page where they can view appointments and edit/delete appointments under “Action”

The “Add appointments” button directs the user to the following page where they can add appointments with patient id.

localhost:3000/doctor_appointment

HOSPITAL MANAGEMENT SYSTEM

Profile

Doctor's HomePage

Appointments

Search

+ Add Appointment

| Appointment ID | Patient ID | Patient Name | Doctor name | Appointment date | Appointment time | Action |
|----------------|------------|--------------|-------------|------------------|------------------|--------|
| 14 | 2 | Jaskaran | sukhdeep | 11/04/2023 | 1:30 PM | ... |
| 17 | 13 | Alokik | aksh | 16/11/2023 | 2:55 AM | ... |
| 20 | 2 | Jaskaran | sukhdeep | 28/03/2023 | 4:04 PM | ... |
| 21 | 19 | Asaiah | aksh | 12/04/2023 | 4:30 PM | ... |
| 23 | 2 | Jaskaran | love | 12/04/2023 | 5:38 AM | ... |
| 24 | 18 | Omid | sukhansh | 29/03/2023 | 12:25 PM | ... |
| 25 | 20 | Shubham | sukhdeep | 05/04/2023 | 12:30 PM | ... |
| 30 | 2 | Jaskaran | aama | 04/04/2023 | 6:22 PM | ... |

The Doctor's salary page lets the doctor view/search his/her salary

A screenshot of a web browser showing the 'Doctor's Salary' page. The URL in the address bar is 'localhost:3000/doctor_salary'. The page has a red header bar with the 'HOSPITAL MANAGEMENT SYSTEM' logo and a 'Profile' button. On the left, there is a sidebar with links: 'Doctor's HomePage', 'Appointments', 'Doctor's Salary' (which is highlighted in blue), 'Medical Records', 'Cases', and 'Operations to perform'. The main content area is titled 'Doctor's Salary' and contains a search bar with a magnifying glass icon and a 'Search' button. Below the search bar is a table with the following data:

| Salary ID | Amount | Doctor Id |
|-----------|---------|-----------|
| 1001 | 10,001 | 230 |
| 1002 | 111 | 228 |
| 1003 | 1234 | 229 |
| 1004 | 1011010 | 228 |
| 1006 | 10,001 | 228 |

The doctor can view/search medical records on the medical records page. They can also edit/delete medical records using the menu under "more."

The screenshot shows the Doctor's HomePage of a Hospital Management System. On the left, a sidebar menu includes: Doctor's HomePage, Appointments, Doctor's Salary, Medical Records (which is selected and highlighted in grey), Cases, and Operations to perform. The main content area is titled "Medical Records" and contains a table of existing medical records. The table has columns: Record Id, Patient Id, Date/Time, Administrator Id, Case Id, and More (with three dots). The data in the table is as follows:

| Record Id | Patient Id | Date/Time | Administrator Id | Case Id | More |
|-----------|------------|-----------|------------------|---------|------|
| 1111 | 18 | any | 1002 | 2002 | ... |
| 1113 | 13 | a | 1002 | 2002 | ... |
| 1115 | 12 | any | 1002 | 2002 | ... |

A blue button labeled "+ Add Medical Record" is located in the top right corner of the main content area.

In order to add medical records, the doctor can use the “add medical record” button to get directed to the following page.

The screenshot shows the "Add Medical Records" page. The left sidebar is identical to the Doctor's HomePage. The main content area is titled "Add Medical Records" and contains four input fields with validation messages: "Patient id *", "Date/Time *", "Administrator Id *", and "Case Id *". Each field has a sample value (2, any, 1002, 2002) and a red asterisk indicating it is required. Below the fields is a blue button labeled "ADD MEDICAL RECORDS".

Edit medical records under the “more” menu will direct you to the following page where you can edit previous medical records.

The screenshot shows a web browser window with the URL `localhost:3000/d_medical_record/d_edit_record/1113`. The page title is "Edit Medical Records". On the left, there is a sidebar with navigation links: Doctor's HomePage, Appointments, Doctor's Salary, Medical Records (which is selected and highlighted in blue), Cases, and Operations to perform. The main content area has four input fields: "Medical Record Id" (1113), "Patient's Id" (13), "Date/Time" (a), and "Administrator's Id" (1002). Below these fields is a "Case's Id" field containing "2002". At the bottom right is a blue button labeled "EDIT MEDICAL RECORDZ".

Choosing “delete” under the “more” menu for a medical record will direct you to the following page to confirm that you want to make this change.

localhost:3000/d_medical_record/d_delete_record/1115

Doctor's HomePage Appointments Doctor's Salary Medical Records Cases Operations to perform

Delete Medical Record

Confirm Delete for Medical Record Id : 1115

The cases page will display all of the medical cases. You can view/search cases by doctor ID.

localhost:3000/d_case

Doctor's HomePage Appointments Doctor's Salary Medical Records Cases Operations to perform

Cases

+ Add Case

| Case Id | Date | Time | Doctor Id | Information |
|---------|------------|----------|-----------|-------------|
| 2002 | 10/04/2022 | 4:57 AM | 230 | doe |
| 2003 | 31/03/2023 | 5:27 PM | 228 | It is done |
| 2004 | 15/04/2023 | 12:25 PM | 230 | It is done |

You can also add new cases by pressing the add button of the top right corner and it will direct you to the following page.

Add Case

Date: 12/04/2023 Time: 6:46 PM

Doctor Id *: 230

Information *

ADD CASE

Operations page displays all the operations. You can view/ search operations using Doctor ID.

Operations to be performed

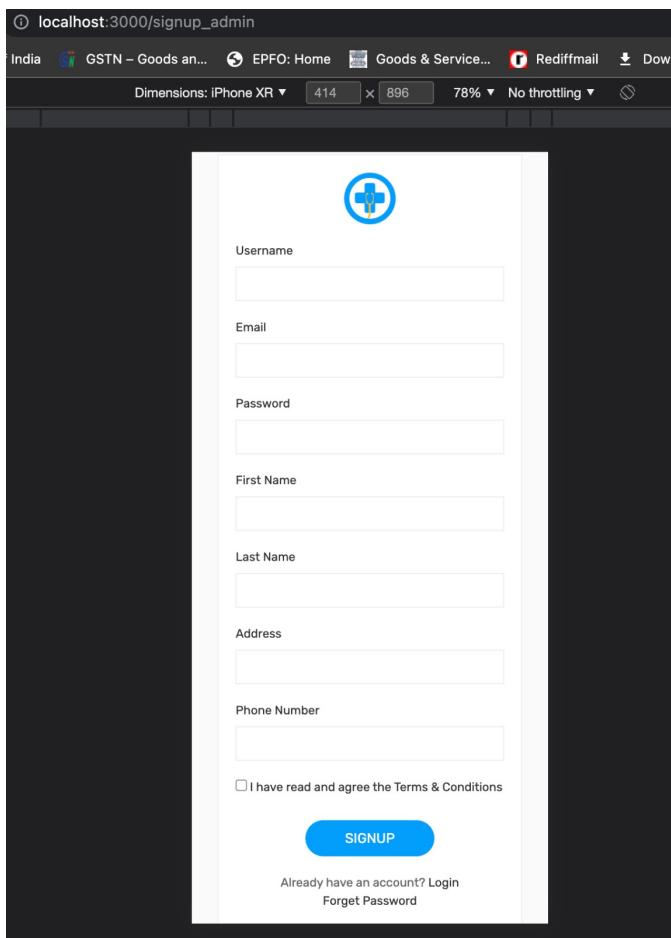
Kindly search for the operation you need to perform using your doctor id:

Search

| Doctor Id | Operation Id | Material Id |
|-----------|--------------|-------------|
| 233 | 3 | 3 |
| 234 | 4 | 10 |
| 235 | 7 | 1 |
| 230 | 8 | 1 |
| 235 | 9 | 3 |

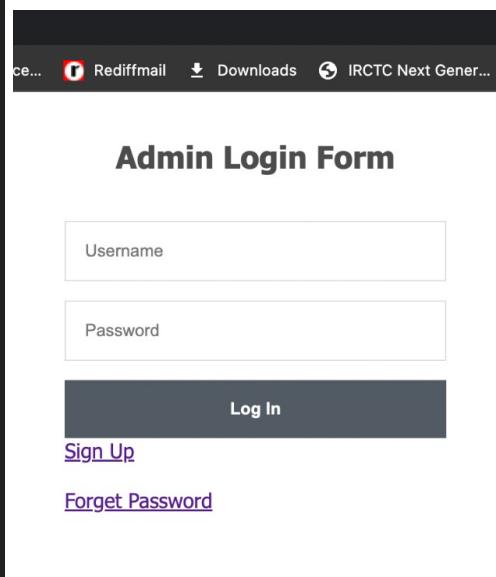
For Admins:

New admins will sign up from selecting “Admin” at the drop down menu if not already added by another Administrator. Existing admins can login directly while new admins need to sign and then login to access their portal. The SignUp/Login Process is pretty simple and similar to other users.



The screenshot shows a web browser window with the URL `localhost:3000/signup_admin`. The page title is "Admin". It features a registration form with fields for Username, Email, Password, First Name, Last Name, Address, and Phone Number. There is also a checkbox for accepting terms and conditions and a "SIGNUP" button. At the bottom, there are links for "Already have an account? Login" and "Forgot Password". The browser's toolbar at the top includes links for India, GSTN – Goods an..., EPFO: Home, Goods & Service..., Rediffmail, and Downloads, along with device dimensions (iPhone XR, 414x896), battery level (78%), and throttling settings.

Admin Sign-Up Page



The screenshot shows a simplified "Admin Login Form" page. It has fields for "Username" and "Password", a "Log In" button, and links for "Sign Up" and "Forgot Password". The page title is "Admin Login Form". The browser's toolbar at the top includes links for Rediffmail, Downloads, and IRCTC Next Gener...

Admin Login Page

Existing admins can go to the login drop down menu and select “Admin” then login
Login page

The homepage will display all the appointments and doctors while the menu on the left will provide the admin with access to different functions regarding doctors, patients, operations, materials, medical records, tests etc. Profile button is on top left where admins can edit profile or logout

EDIT PROFILE PAGE

The screenshot shows the homepage of a Hospital Management System. On the left, there is a sidebar with various menu items: HomePage, Doctors, Administrator, Patients, Appointments, Salary, Operation, Materials Used, Medical Records, Tests, and Cases. The main content area has two tables. The first table, titled "8 Appointments", lists patient names, appointment times, and doctors. The second table, titled "Doctors", lists doctor names, MBBS status, and their profiles.

| 8 Appointments | | |
|--------------------------|------------------------------|--------------------|
| Patient Name Jaskaran | Appointment With Sukhansh | Timing 1:30 PM |
| Patient Name Alokik | Appointment With Heer | Timing 2:55 AM |
| Patient Name Jaskaran | Appointment With Aron | Timing 4:04 PM |
| Patient Name Asaiah | Appointment With Aron | Timing 4:30 PM |
| Patient Name Jaskaran | Appointment With Aksh | Timing 5:38 AM |
| Patient Name Omid | Appointment With Sukhdeep | Timing 12:25 PM |
| Patient Name Shubham | Appointment With Shubbh | Timing 12:30 PM |
| Patient Name Jaskaran | Appointment With Sukhansh | Timing 6:22 PM |

| Doctors | |
|----------------------------|----------|
| Aksh Singh MBBS, MD | Show all |
| Sukhansh Nagi MBBS, MD | |
| Sukhdeep Arora MBBS, MD | |
| Aron Midge MBBS, MD | |
| Heer Jeanie | |

The Doctors page provides a list of all doctors. The admin can view/search doctors.

The screenshot shows a web application interface for managing doctors. On the left, there's a sidebar with various menu items like HomePage, Doctors, Administrator, Patients, Appointments, Salary, Operation, Materials Used, Medical Records, Tests, and Cases. The main content area is titled 'Doctors' and contains a table with the following data:

| ID | username | First Name | Last Name | Email | Address | Phone | More |
|-----|----------|------------|-----------|-----------------|----------|-------------|------|
| 228 | aksh | Aksh | Singh | aksh@gmail.com | 12st | 58796641111 | |
| 229 | user1 | Sukhansh | Nagi | b@b.com | gt | 567 | |
| 230 | user2 | Sukhdeep | Arora | sukh@gmail.com | 12st | 58655 | |
| 233 | aam | Aron | Midge | amam@ama.comm | ama23223 | 22309123 | |
| 234 | mr love | Heer | Jeanie | jihad@gmail.com | 12st | 1233 | |
| 235 | shubh | Shubh | Charles | shubb@gmail.com | 12we | 1233 | |

The admin can add new doctors using the “Add Doctor” button on the top right corner.

The screenshot shows the 'Add Doctor' form. The sidebar on the left remains the same. The main form has the following fields:

- Username *: mmd
- Password *: (represented by dots)
- First Name *: Muhammad
- Last Name *: Abbas
- Email *: abbas@gmail.com
- Address: 12 Vincent Road
- Phone: 5669844931

At the bottom right of the form is a blue 'CREATE DOCTOR' button.

The admins can also edit/delete doctors using the menu under “more”

localhost:3000/doctors/edit_doctor/228#

HomePage

Doctors

Administrator

Patients

Appointments

Salary

Operation

Materials Used

Medical Records

Tests

Cases

Edit Doctor

Doctor ID *

228

Password

First Name *

Aksh

Last Name

Singh

Email Address

aksh@gmail.com

Address

12st

Phone

58796641111

EDIT DOCTOR

localhost:3000/doctors/delete_doctor/228

HomePage

Doctors

Administrator

Patients

Appointments

Salary

Operation

Materials Used

Medical Records

Tests

Cases

Delete Doctor

Confirm Delete for doctor id : 228

On the administrator page you can view/search administrators.

localhost:3000/administrator

HOSPITAL MANAGEMENT SYSTEM

Administrator

+ Add Administrator

| ID | Username | Email | Password | First Name | Last Name | Address | Phone Number | More |
|------|----------|-------------------------|------------|------------|-------------------|--------------------------|--------------|------|
| 1001 | mo | mo@mo.co | 12 | mo | momo | momomo | 2233322 | |
| 1002 | jsb | alokisksingh2@gmail.com | 123 | g | f | f | 55885 | |
| 1003 | jayesssb | jskrnbhatia@gmail.com | Qazwsx@255 | jaskaran | Bhatia | 4-2704 brentwood blvd nw | 5879663848 | |
| 1004 | zihad | h@gmail.com | 123 | hh | hh | 7 | 78888 | |
| 1005 | shivam | alokisksingh2@gmail.com | 123 | shivam | jagdev | 12st Aspen Woods | 1234567 | |
| 1006 | rohil | abcd@gmail.com | rohil | Rohil | Dhillon | 123 chippar drive | 123456891 | |
| 1007 | rohil | 12345 | rohil | dhillon | dhillon@gmail.com | 12st | 12345 | |

An admin can also edit/delete administrators from the option under the “more” menu.

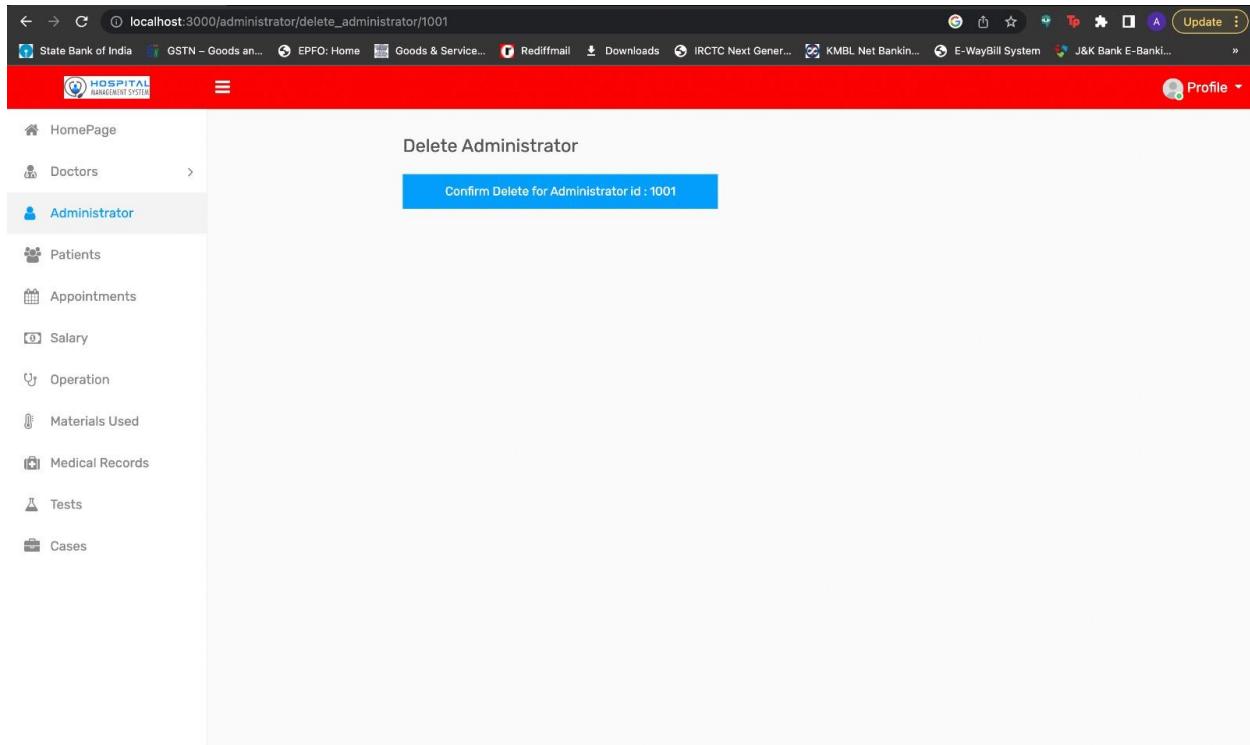
localhost:3000/administrator/edit_administrator/1001

HOSPITAL MANAGEMENT SYSTEM

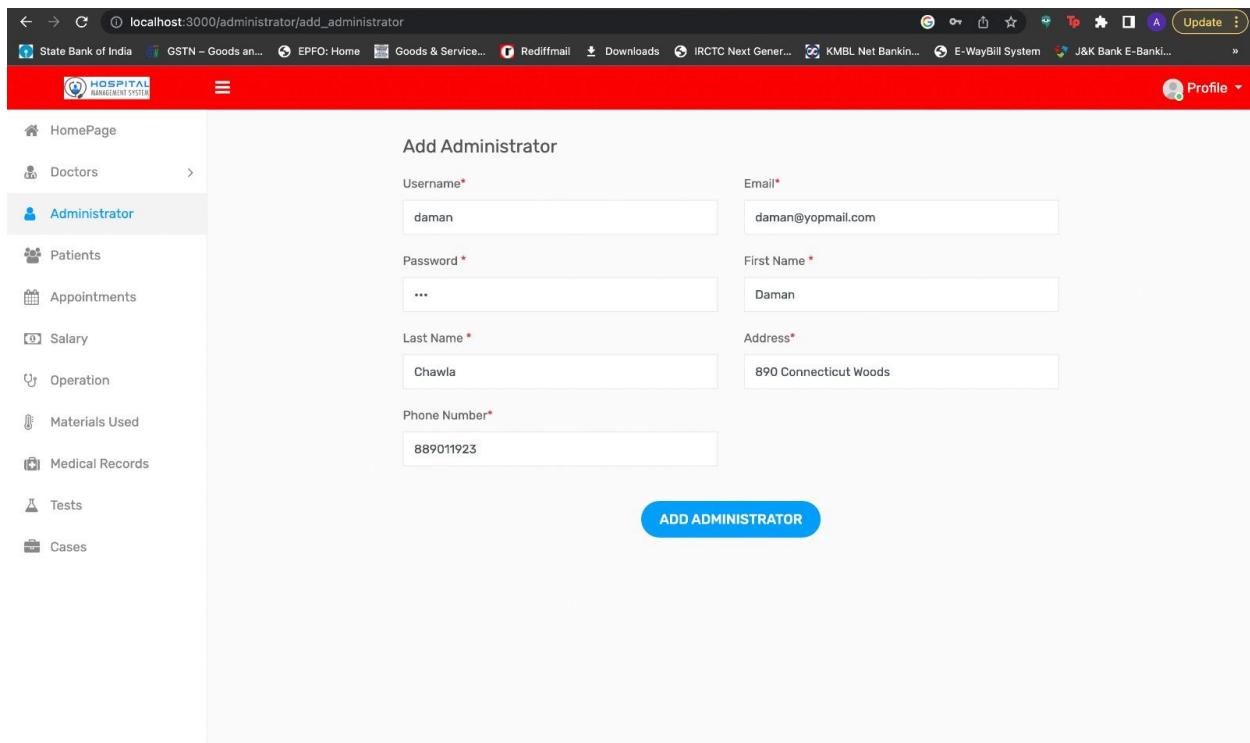
Edit Administrator

| | |
|------------------|--------------|
| Administrator id | Username |
| 1001 | mo |
| Email | Password |
| mo@mo.co | .. |
| First Name | Last Name |
| mo | mo |
| Address | Phone Number |
| 13 Street | 2233322 |

UPDATE ADMINISTRATOR



An admin can also add new admins.



On the patients page, an admin can view/search patients.

localhost:3000/patients

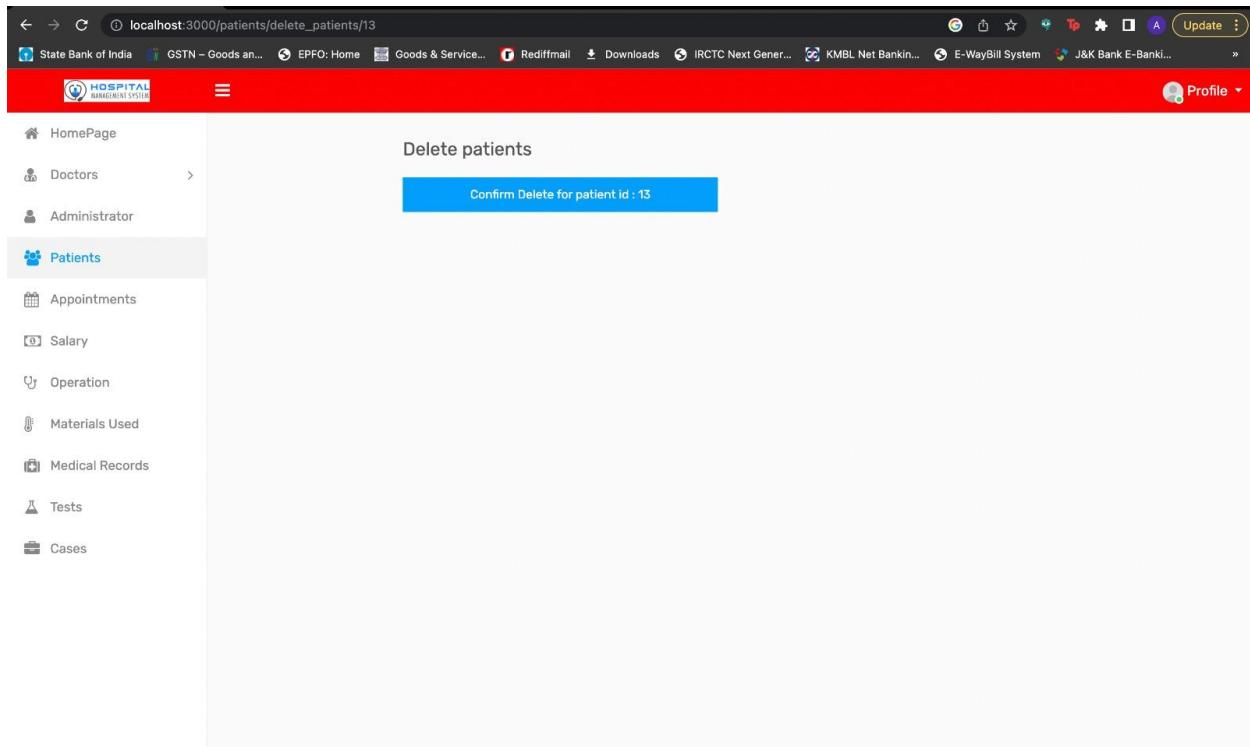
The screenshot shows a web-based hospital management system. The top navigation bar includes links for State Bank of India, GSTN – Goods an..., EPFO: Home, Goods & Service..., Rediffmail, Downloads, IRCTC Next Gener..., KMBL Net Bankin..., E-WayBill System, J&K Bank E-Banki..., and a Profile dropdown. The main content area has a red header "PATIENTS". On the left, a sidebar menu lists: HomePage, Doctors, Administrator, Patients (selected), Appointments, Salary, Operation, Materials Used, Medical Records, Tests, and Cases. The main table displays patient data with columns: ID, Username, Password, First Name, Last Name, Email, Address, Phone, and More (with three dots). The data is as follows:

| ID | Username | Password | First Name | Last Name | Email | Address | Phone | More |
|----|----------|----------|------------|-----------|-------------------|----------------------------|------------|------|
| 2 | jaskaran | 123 | Jaskaran | j | j@gmail.com | i | 877 | ... |
| 4 | rohil | jaskaran | jasaran | b | b@b.com | 12st | 51234 | ... |
| 12 | ASD | yty | Lin | Yin | rt@gmail.com | 12st | 56799882 | ... |
| 13 | sukh | 123 | Sukh | Singh | ao10@gmail.com | #450 st | 235509992 | ... |
| 18 | omid | 123 | omid | badaghi | omid@ucalgary.ca | 12st | 123 | ... |
| 19 | Asaiah | asaiah | Asaiah | Zeinab | asaiah@gmail.com | 6th Avenue, Spring Village | 5878779001 | ... |
| 20 | shubham | 123 | Shubham | Aggarwal | shubham@gmail.com | 12 St | 576848222 | ... |

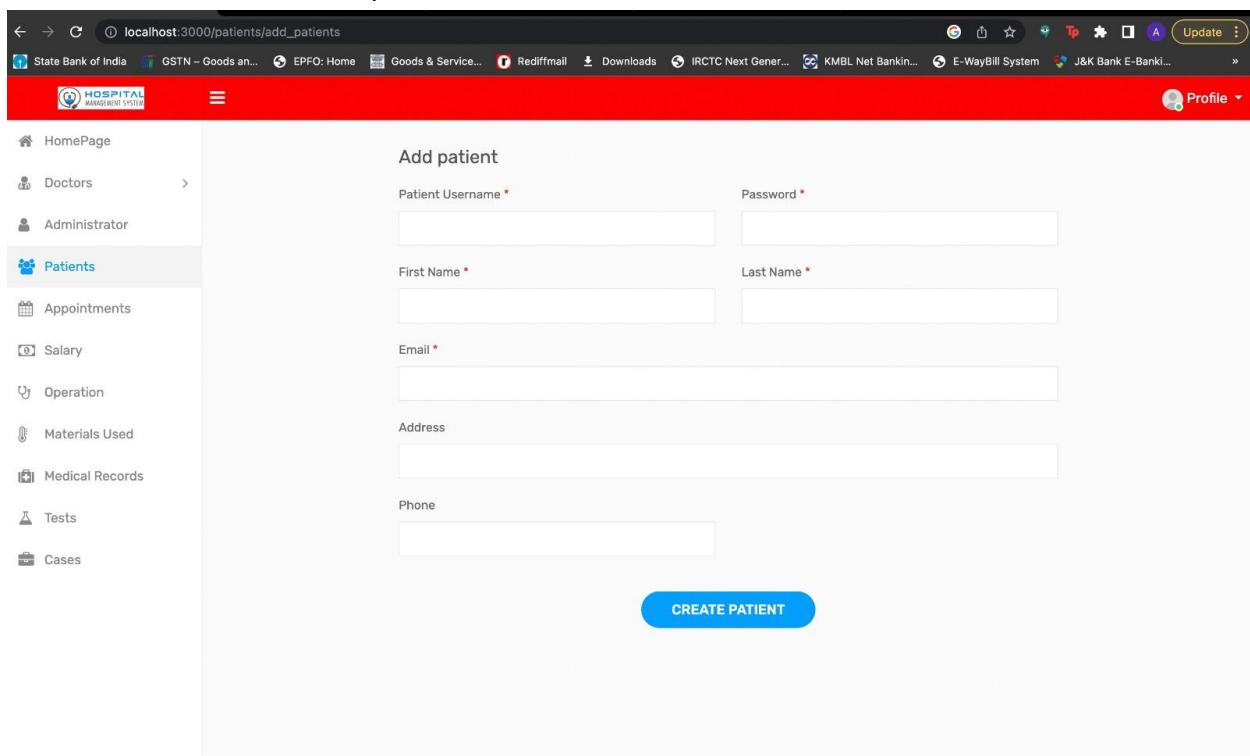
An admin can also edit/delete patients from the menu under “more.”

localhost:3000/patients/edit_patients/2

The screenshot shows the "Edit Patients" form. The top navigation bar and sidebar are identical to the previous screenshot. The main form has fields for Patient ID * (2), Username (jaskaran), Password (***), First Name * (Jaskaran), Last Name (j), Email Address (j@gmail.com), Address (i), and Phone (877). A large blue button at the bottom right says "EDIT PATIENT".



An admin can also add new patients.



The appointments page shows all the appointments. An admin can view/search , add and edit/delete appointments.

| Appointment ID | Patient ID | Patient Name | Doctor name | Appointment date | Appointment time | More |
|----------------|------------|--------------|-------------|------------------|------------------|------|
| 14 | 2 | Jaskaran | Sukhansh | 11/04/2023 | 1:30 PM | ... |
| 17 | 2 | Alokik | Heer | 16/11/2023 | 2:55 AM | ... |
| 20 | 2 | Jaskaran | Aron | 28/03/2023 | 4:04 PM | ... |
| 21 | 2 | Asaiah | Aron | 12/04/2023 | 4:30 PM | ... |
| 23 | 2 | Jaskaran | Aksh | 12/04/2023 | 5:38 AM | ... |
| 24 | 2 | Ornid | Sukhdeep | 29/03/2023 | 12:25 PM | ... |
| 25 | 2 | Shubham | Shubh | 05/04/2023 | 12:30 PM | ... |
| 30 | 2 | Jaskaran | Sukhansh | 04/04/2023 | 6:22 PM | ... |

The salary page shows the Doctor's salary. The admin can view/ search the doctors' salary.

| Salary ID | Amount | Doctor Id | More |
|-----------|---------|-----------|------|
| 1001 | 10,001 | 230 | ... |
| 1002 | 111 | 228 | ... |
| 1003 | 1234 | 229 | ... |
| 1004 | 1011010 | 228 | ... |
| 1006 | 10,001 | 228 | ... |

An admin can also edit/delete salaries from the menu under “more”

localhost:3000/salary/edit_salary/1001

HOSPITAL MANAGEMENT SYSTEM

Profile

HomePage Doctors Administrator Patients Appointments Salary Operation Materials Used Medical Records Tests Cases

Edit salary

| | |
|-----------|--------|
| salary ID | Amount |
| 1001 | 10,000 |

Doctor ID: 230

EDIT SALARY

localhost:3000/salary/delete_salary/1003

HOSPITAL MANAGEMENT SYSTEM

Profile

HomePage Doctors Administrator Patients Appointments Salary Operation Materials Used Medical Records Tests Cases

Delete Salary

Confirm Delete for Salary Id : 1003

An admin can also add new salaries using the “add salary” button.

localhost:3000/salary/add_salary

Add Salary

Amount *

Doctor Id *

ADD SALARY

Navigation menu:

- HomePage
- Doctors
- Administrator
- Patients
- Appointments
- Salary**
- Operation
- Materials Used
- Medical Records
- Tests
- Cases

The operations page shows the operations performed by doctors. An admin can view/search operations.

localhost:3000/operation

Operations performed by doctors

+ Add operation

Search

| Operation Id | Name of Operation | Cost | Patient id | Doctor id | Material Id | More |
|--------------|-------------------|-------|------------|-----------|-------------|------|
| 3 | Leg | 2 | 2 | 233 | 3 | ... |
| 4 | Ear | 10000 | 12 | 234 | 10 | ... |
| 7 | Heart | 109 | 4 | 235 | 1 | ... |
| 8 | Leg | 3 | 13 | 230 | 1 | ... |
| 9 | Stomach | 10000 | 13 | 235 | 3 | ... |

An admin can also edit/delete operations using the menu under “more”

The screenshot shows the 'Edit operation' page of a hospital management system. The URL is localhost:3000/operation/edit_operation/3. The left sidebar has a 'Operation' section selected. The main form contains the following fields:

| | |
|--------------|------------|
| Operation ID | Name |
| 3 | Select |
| Cost | Patient ID |
| 200 | 2 |
| Doctor ID | |
| 233 | |
| Material ID | |
| 3 | |

EDIT OPERATION button at the bottom.

The screenshot shows the 'Delete operation' confirmation page of a hospital management system. The URL is localhost:3000/operation/delete_operation/7. The left sidebar has a 'Operation' section selected. A blue button in the center says 'Confirm Delete for Material Id : 7'.

An admin can also add new operations using “add operations” button.

localhost:3000/operation/add_operation

HOSPITAL MANAGEMENT SYSTEM

Add operation

Operation name* Brain

Cost* 10000

Patient id*

Doctor Id* Material Id*

ADD OPERATION

HomePage Doctors Administrator Patients Appointments Salary **Operation** Materials Used Medical Records Tests Cases

The materials used page helps the admin keep track of materials used in operations. An admin can view/search materials

localhost:3000/materials

HOSPITAL MANAGEMENT SYSTEM

Materials Used in Operations

+ Add Materials

| ID | Name of material | Cost | Appointment id | Operation id | More |
|----|------------------|------|----------------|--------------|------|
| 1 | Scalpel | 100 | 2 | 3 | ... |
| 3 | Bandage | 23 | 2 | 3 | ... |
| 10 | Needle | 5 | 12 | 3 | ... |

Search

HomePage Doctors Administrator Patients Appointments Salary Operation **Materials Used** Medical Records Tests Cases

Admin can add/ edit/ delete materials accordingly.

localhost:3000/materials/add_materials

Add Materials

Name *

Select

Cost *

Appointment id

Operation Id

ADD MATERIALS

HomePage

Doctors

Administrator

Patients

Appointments

Salary

Operation

Materials Used

Medical Records

Tests

Cases

localhost:3000/operation/edit_operation/3

Edit operation

Operation ID

3

Name

Select

Cost

200

Patient ID

2

Doctor ID

233

Material ID

3

EDIT OPERATION

HomePage

Doctors

Administrator

Patients

Appointments

Salary

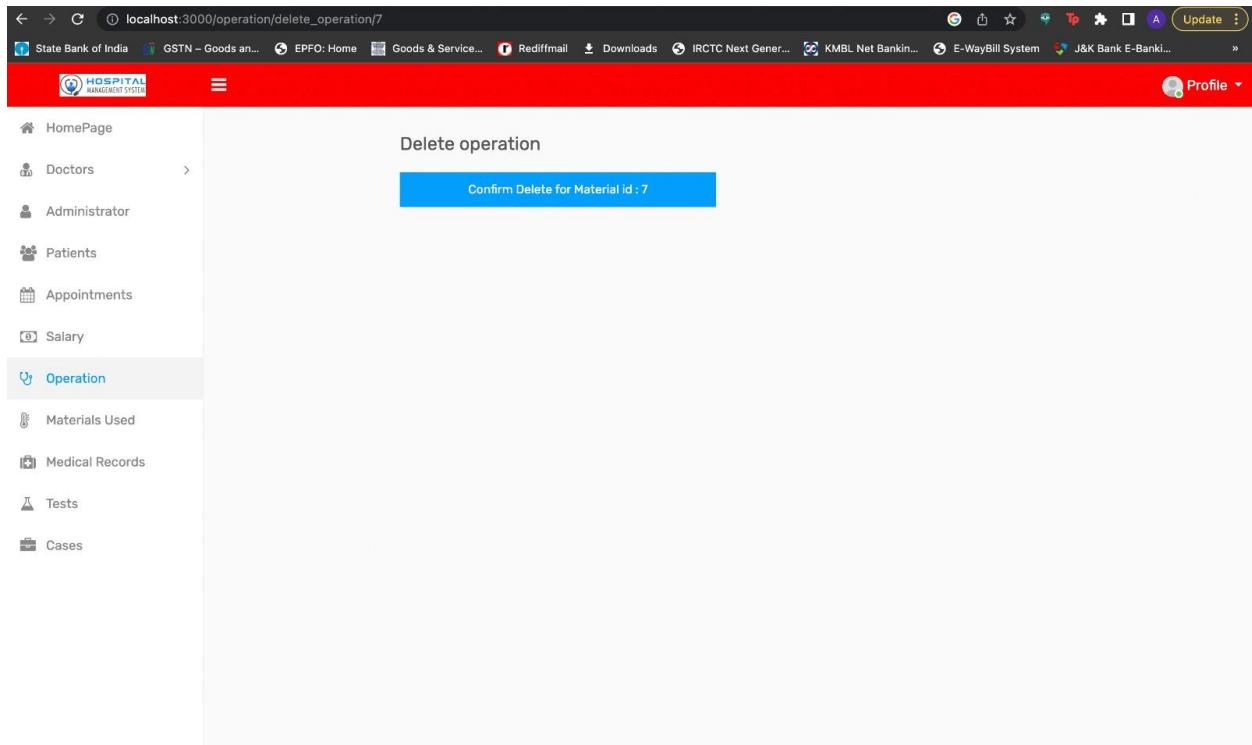
Operation

Materials Used

Medical Records

Tests

Cases



The medical records page displays all the medical records. An admin can view/search for medical records. Additionally, they can also edit/delete medical records and add new medical records.

localhost:3000/medical_record

HOSPITAL MANAGEMENT SYSTEM

Profile

Medical Records

+ Add Medical Record

Record Id Patient Id Date/Time Administrator Id Case Id More

| | | | | | |
|------|----|-----|------|------|-----|
| 1111 | 18 | any | 1002 | 2002 | ... |
| 1113 | 13 | a | 1002 | 2002 | ... |
| 1115 | 12 | any | 1002 | 2002 | ... |

Search

HomePage Doctors Administrator Patients Appointments Salary Operation Materials Used Medical Records Tests Cases

localhost:3000/medical_record/add_record

HOSPITAL MANAGEMENT SYSTEM

Profile

Add Medical Records

Patient id *

Date/Time *

Administrator Id *

Case Id *

ADD MEDICAL RECORDS

HomePage Doctors Administrator Patients Appointments Salary Operation Materials Used Medical Records Tests Cases

localhost:3000/medical_record/edit_record/1118

HOSPITAL MANAGEMENT SYSTEM

Edit Medical Records

Medical Record Id: 1118

Patient's Id: 19

Date/Time:

Administrator's Id: 1002

Case's Id: 2002

EDIT MEDICAL RECORDZ

localhost:3000/medical_record/delete_record/1115

HOSPITAL MANAGEMENT SYSTEM

Delete Medical Record

Confirm Delete for Medical Record Id : 1115

The Lab tests page displays all the test results. An admin can view/search for test results, they can also add, edit, delete lab tests accordingly.

localhost:3000/lab_test

State Bank of India GSTN – Goods an... EPFO: Home Goods & Service... Rediffmail Downloads IRCTC Next Gener... KMBL Net Bankin... E-WayBill System J&K Bank E-Banki... Profile

Lab tests

+ Add Lab Test

| Test Id | Record Id | Type of Test | More |
|---------|-----------|--------------|------|
| 190 | 1111 | Ultrasound | ... |
| 191 | 1113 | Biopsy | ... |
| 192 | 1111 | Ultrasound | ... |
| 194 | 1111 | X-Ray | ... |
| 195 | 1113 | Urine Test | ... |

Search

HomePage Doctors Administrator Patients Appointments Salary Operation Materials Used Medical Records Tests Cases

localhost:3000/lab_test/edit_test/190

State Bank of India GSTN – Goods an... EPFO: Home Goods & Service... Rediffmail Downloads IRCTC Next Gener... KMBL Net Bankin... E-WayBill System J&K Bank E-Banki... Profile

Edit Lab Test

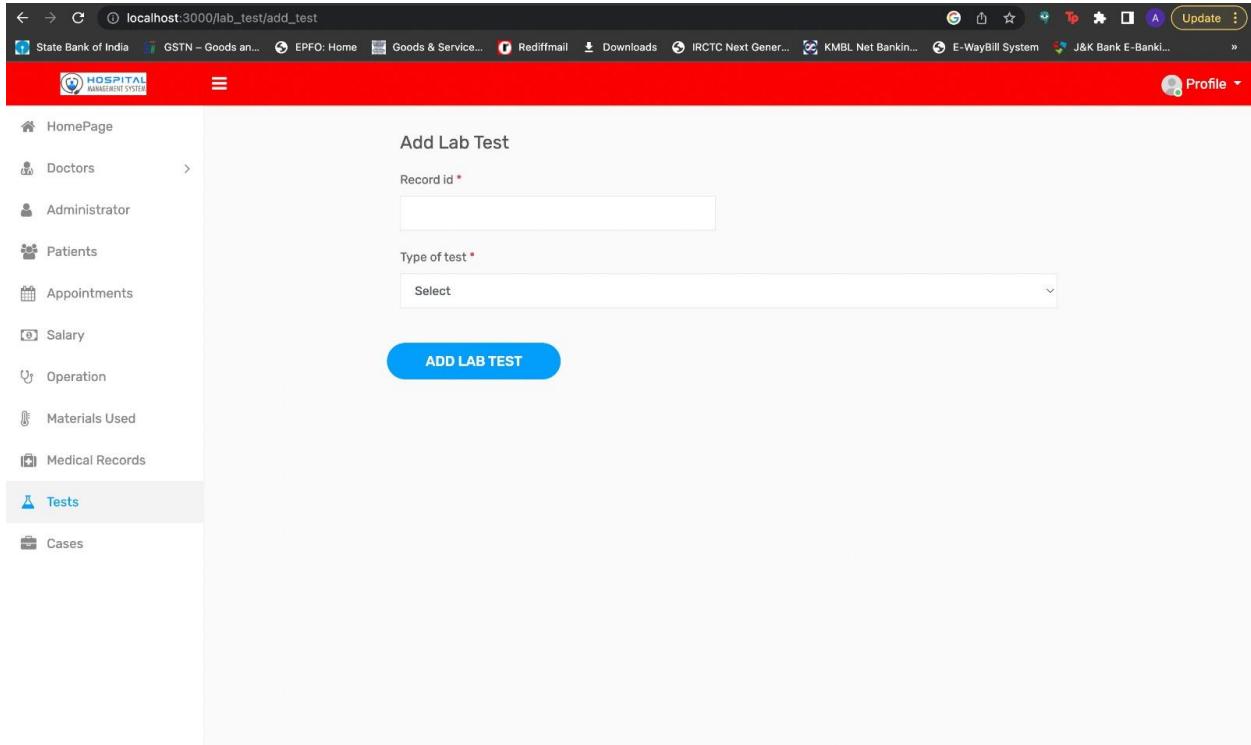
Lab Test Id: 190 Record's Id: 1111

Type of test *

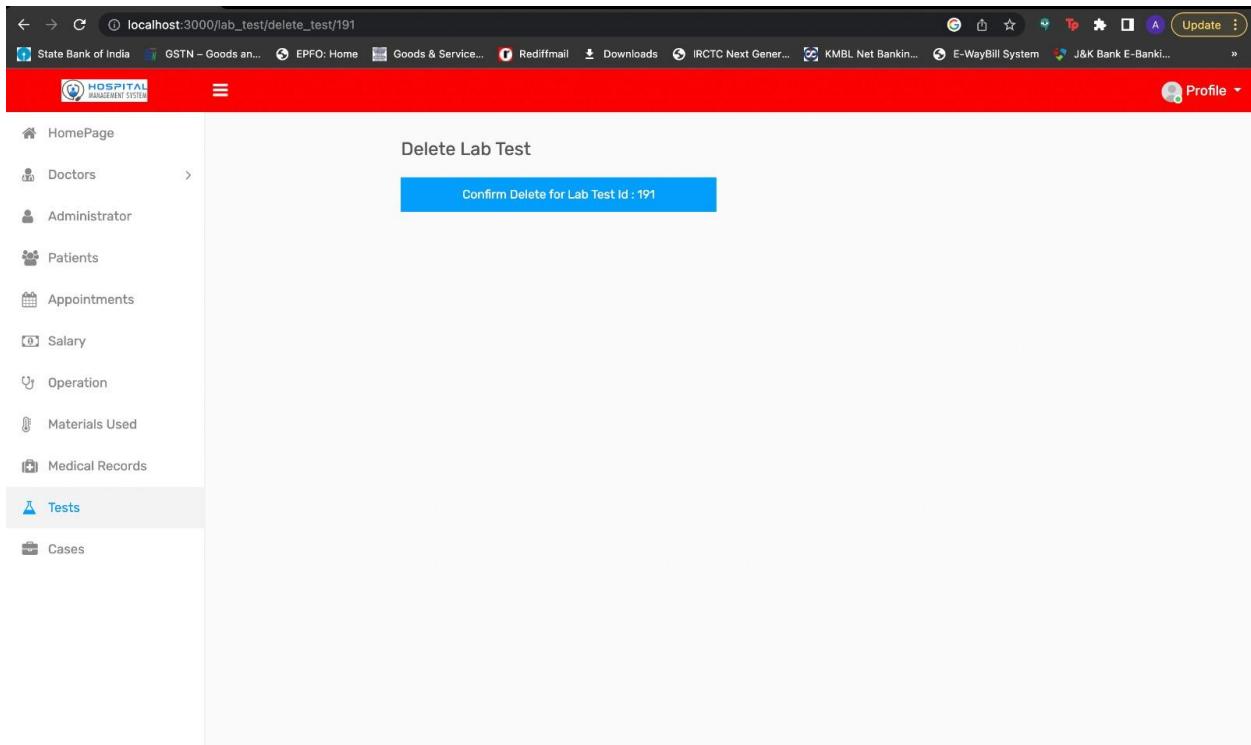
Select

EDIT LAB TEST

HomePage Doctors Administrator Patients Appointments Salary Operation Materials Used Medical Records Tests Cases



The screenshot shows a web browser window with the URL `localhost:3000/lab_test/add_test`. The title bar includes the browser's address bar, a back/forward button, and an update icon. The main content area has a red header bar with the hospital logo and a search bar labeled "Profile". On the left, a sidebar menu lists various hospital management functions: HomePage, Doctors, Administrator, Patients, Appointments, Salary, Operation, Materials Used, Medical Records, Tests (which is selected and highlighted in blue), and Cases. The main content area is titled "Add Lab Test" and contains two input fields: "Record id *" and "Type of test *". Below these is a blue "ADD LAB TEST" button.



The screenshot shows a web browser window with the URL `localhost:3000/lab_test/delete_test/191`. The layout is identical to the previous screenshot, with the same red header, sidebar menu, and "Tests" being the active tab. The main content area is titled "Delete Lab Test" and features a blue button with the text "Confirm Delete for Lab Test Id : 191".

The cases page displays all the cases with their corresponding dates. An admin can view/search for cases, they can also add new cases and edit or delete previous cases.

localhost:3000/cases

HOSPITAL MANAGEMENT SYSTEM

Profile

HomePage

Doctors

Administrator

Patients

Appointments

Salary

Operation

Materials Used

Medical Records

Tests

Cases

Cases

+ Add Case

Search

| Case Id | Date | Time | Doctor Id | Information | More |
|---------|------------|---------|-----------|-------------|------|
| 2002 | 28/03/2023 | 3:57 AM | 230 | It is done | ... |
| 2007 | 31/03/2023 | 7:02 PM | 229 | It is done | ... |

localhost:3000/cases/add_case

HOSPITAL MANAGEMENT SYSTEM

Profile

HomePage

Doctors

Administrator

Patients

Appointments

Salary

Operation

Materials Used

Medical Records

Tests

Cases

Add Case

Date

Time

Doctor Id

Information

CREATE CASE

localhost:3000/cases/add_case

HOSPITAL MANAGEMENT SYSTEM

Profile

HomePage

Doctors

Administrator

Patients

Appointments

Salary

Operation

Materials Used

Medical Records

Tests

Cases

Add Case

Date

Time

Doctor Id

Information

CREATE CASE

localhost:3000/cases/edit_case/2002

State Bank of India GSTN – Goods an... EPFO: Home Goods & Service... Rediffmail Downloads IRCTC Next Gener... KMBL Net Bankin... E-WayBill System J&K Bank E-Banki... Update

HOSPITAL MANAGEMENT SYSTEM Profile

HomePage Doctors Administrator Patients Appointments Salary Operation Materials Used Medical Records Tests Cases

Edit Case

| | |
|----------------------|----------------------|
| Case ID | Date |
| 2002 | <input type="text"/> |
| Time | Doctor Id |
| <input type="text"/> | 230 |
| Information | |
| It is done | |

EDIT CASE

localhost:3000/cases/delete_case/2002

State Bank of India GSTN – Goods an... EPFO: Home Goods & Service... Rediffmail Downloads IRCTC Next Gener... KMBL Net Bankin... E-WayBill System J&K Bank E-Banki... Update

HOSPITAL MANAGEMENT SYSTEM Profile

HomePage Doctors Administrator Patients Appointments Salary Operation Materials Used Medical Records Tests Cases

Delete Case

Confirm Delete for case id : 2002

