**New Jersey Institute of Technology**

# CS631 Database System Design Final Project Report

## "NEWARK MEDICAL ASSOCIATION'S HOSPITAL MANAGEMENT SYSTEM"

**Implemented By:**

**Dinesh Lankepillewar**

**Ojaswi Ghate**

**Guided by:**

**Prof. Charles Yoo**

# Content Table

## 1. Objective & Goals

The objective is to prepare an application for hospital management, which could maintain data and provide a user-friendly interface for retrieving patients and also hospital related details efficiently and accurately.

Goal is to manage the data related to hospital's departments such as,

- Patient Management
- In-patient Management
- In medical staff management

For Example,

**Patient data retrieval:**

Hospital Management System application makes it possible to access all the data related to a patient via a system by the means of a few simple clicks. Information like patient history, current illness, doctors involved can be made visible to the user. These data will help to connect the dots about the patient, like specific diagnosis.

## 2. Introduction

Hospital management system application is a web based application. This technical report describes our design for a database system to be used by a hospital to keep daily record for Hospital management.

Hospital management systemapplication is introduced with the cause for helping hospital speed up their processes.Hospital management system is used to solve the complications coming from managing all the paper works associated with the various departments of hospitalization with confidentiality. For example, itprovides the ability to reduce staff work in arranging and analyzing the paperwork of the patients.Hospital management system do works like,

- Maintain the medical records of the patients
- Keep the track of the appointment dates
- Keep the track of scheduled appoints etc
- Register new patient with his details

### 2.1 Advantages of Hospital management system application

**The advantages of Hospital management system application can be pinpointed to the following:**

- Time-saving Technology
- `Improved Efficiency by avoiding human errors and also reduces scope for Errors
- Data security and correct data retrieval made possible
- Easy access to patient data with correct patient history

## 3.Summary of System Requirements

In this technical project, the database system design is used as a medium tool in Hospital system design management. This project requires the hospital system to retrieve information of its patients, doctors, and staff.

### 3.1 System Requirements:

- **Patient Management Application**

  Provide interface for employee to:

  - Register new patient
  - View patient details
  - Select an appointment with a doctor
  - Check previous diagnoses and illness
  - View scheduled per doctor per day

- **In-patient Management**

  Provide interface to employee to:

  - Check for available room/bed
  - Assign/remove a room to more patients
  - Assign/remove a doctor to more patients
  - Assign/remove a nurse to more patients
  - View scheduled per doctor per day
  - View scheduled surgery per room per day
  - Book a surgery
  - View scheduled surgery per patient

- **In medical staff management**
  Provide interface to employee to:

  - Add staff member
  - Delete staff member
  - Schedule job shift
  - View staff member per job type

## 3.2 Application Implementation

- Utilized HTML, Core Java, JSP, Servlets, JDBC, MySQL to implement our application.
- JDBC (Java Database connectivity) helped to connect to our database.
- For implementing the front end, used HTML and JSP Servlets.
- Apache Tomcat server to deploy the application.
- MySQL Workbench is used to manage database and organize data in databases.

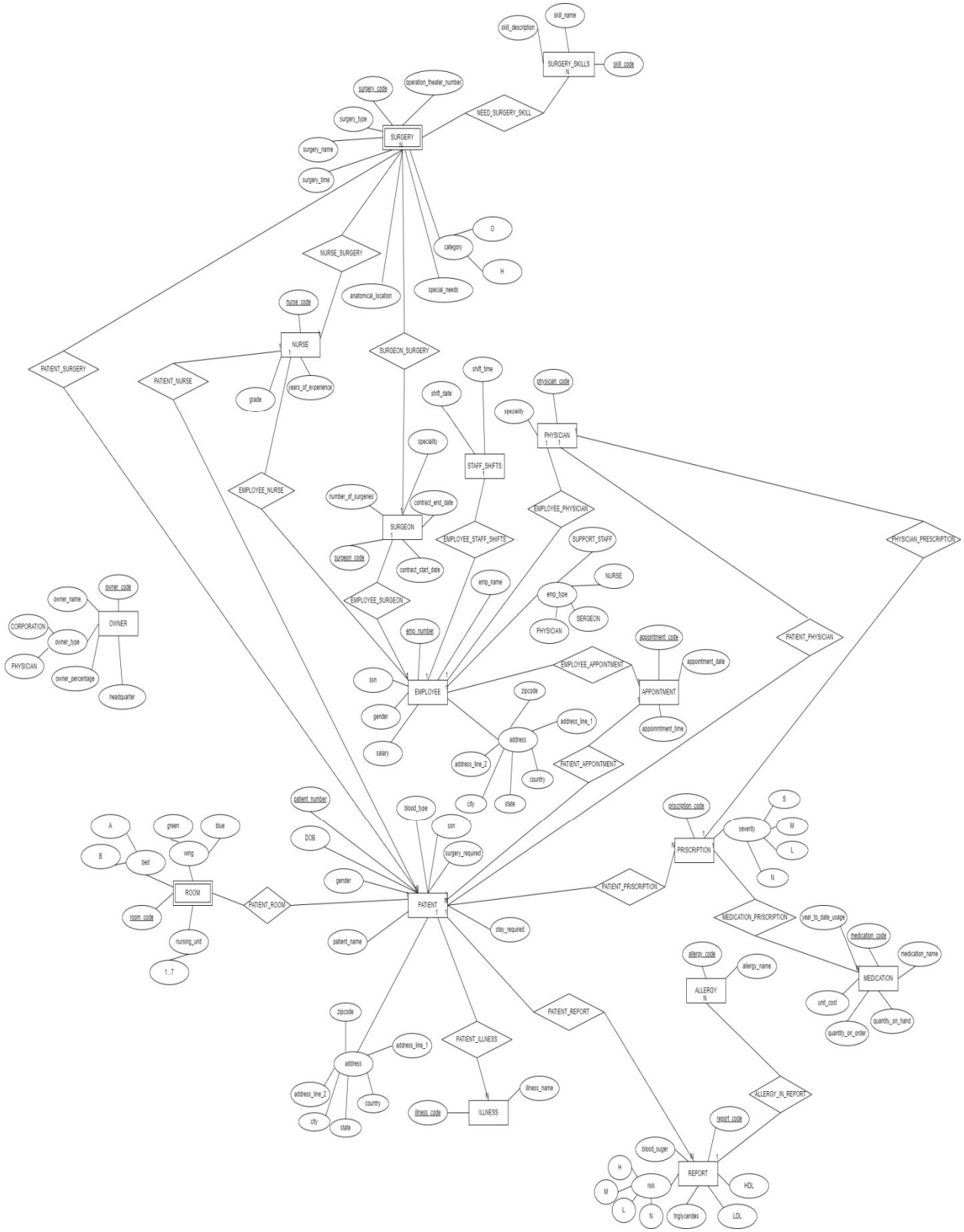
## 4. Design Process

System Design consists of two parts: Database Design and Application Design.

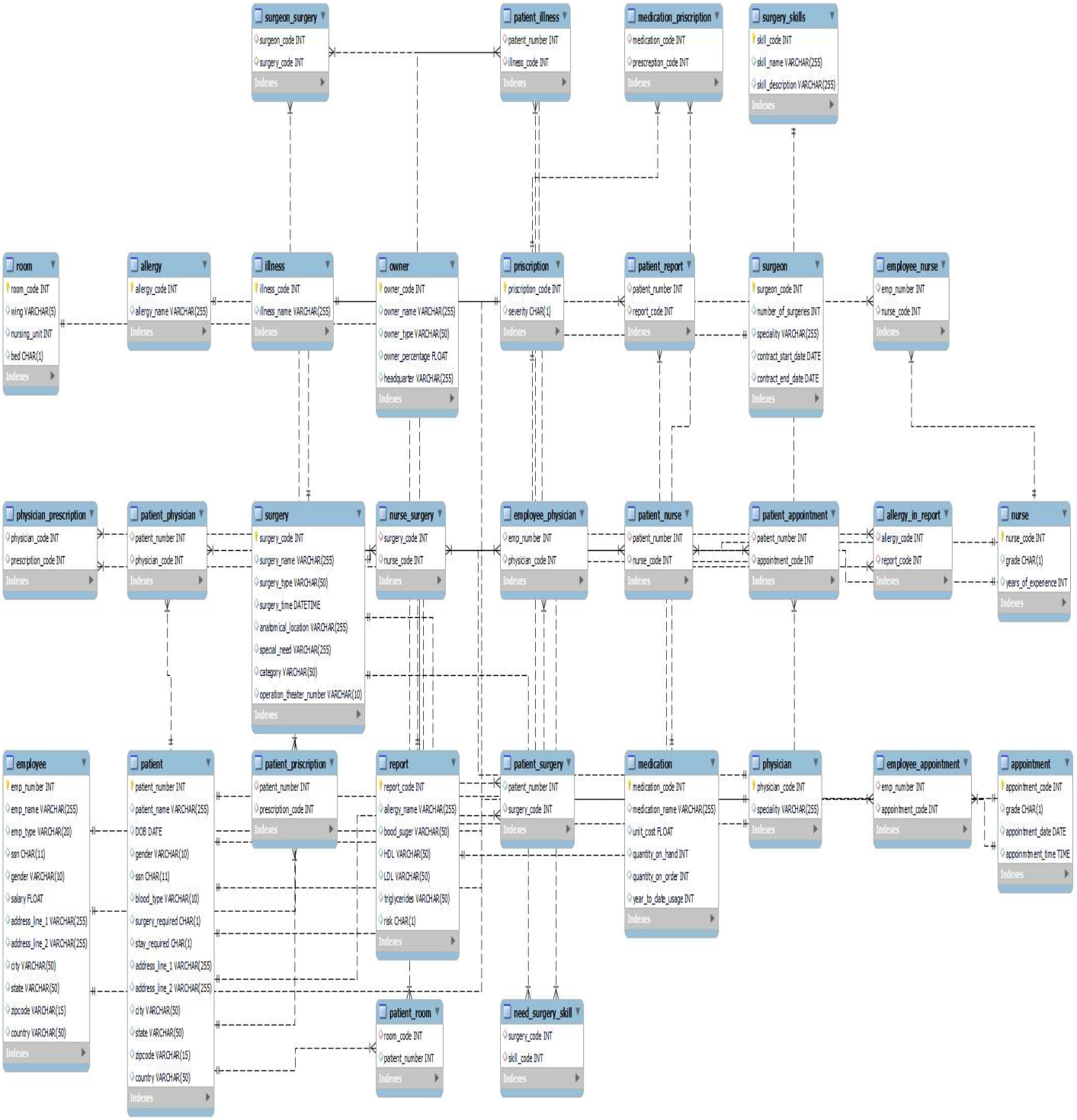
## 4.1 Database Design :

This consist of Entity –Relationship diagram, Relational Schema, Table creation SQL queries.

# Entity –Relationship diagram



## Relational schema design



## Relation Schema:

```
-- MySQL Script generated by MySQL Workbench

-- Wed Nov 30 22:53:05 2022

-- Model: New Model    Version: 1.0

-- MySQL Workbench Forward Engineering


SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,
NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';


-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

SHOW WARNINGS;

-- -----------------------------------------------------

-- Schema newarkma

-- -----------------------------------------------------
```

```
DROP SCHEMA IF EXISTS `newarkma` ;


-- -----------------------------------------------------
-- Schema newarkma
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `newarkma` DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_0900_ai_ci ;
SHOW WARNINGS;
USE `newarkma` ;


-- -----------------------------------------------------
-- Table `allergy`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `allergy` ;


SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `allergy` (
  `allergy_code` INT NOT NULL,
  `allergy_name` VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (`allergy_code`))
ENGINE = InnoDB
```

```
DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;


-- -----------------------------------------------------

-- Table `allergy_in_report`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `allergy_in_report` ;


SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `allergy_in_report` (

  `allergy_code` INT NULL DEFAULT NULL,

  `report_code` INT NULL DEFAULT NULL)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;
```

```
-- -----------------------------------------------------
-- Table `appointment`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `appointment` ;


SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `appointment` (
  `appointment_code` INT NOT NULL,
  `grade` CHAR(1) NULL DEFAULT NULL,
  `appointment_date` DATE NULL DEFAULT NULL,
  `appoinmtment_time` TIME NULL DEFAULT NULL,
  PRIMARY KEY (`appointment_code`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;


-- -----------------------------------------------------
-- Table `employee`
```

```
-- -----------------------------------------------------

DROP TABLE IF EXISTS `employee` ;


SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `employee` (

  `emp_number` INT NOT NULL AUTO_INCREMENT,

  `emp_name` VARCHAR(255) NULL DEFAULT NULL,

  `emp_type` VARCHAR(20) NULL DEFAULT NULL,

  `ssn` CHAR(11) NULL DEFAULT NULL,

  `gender` VARCHAR(10) NULL DEFAULT NULL,

  `salary` FLOAT NULL DEFAULT NULL,

  `address_line_1` VARCHAR(255) NULL DEFAULT NULL,

  `address_line_2` VARCHAR(255) NULL DEFAULT NULL,

  `city` VARCHAR(50) NULL DEFAULT NULL,

  `state` VARCHAR(50) NULL DEFAULT NULL,

  `zipcode` VARCHAR(15) NULL DEFAULT NULL,

  `country` VARCHAR(50) NULL DEFAULT NULL,

  PRIMARY KEY (`emp_number`))

ENGINE = InnoDB

AUTO_INCREMENT = 5
```

```
DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;


-- -----------------------------------------------------

-- Table `employee_appointment`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `employee_appointment` ;


SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `employee_appointment` (

  `emp_number` INT NULL DEFAULT NULL,

  `appointment_code` INT NULL DEFAULT NULL)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;
```

```
-- -----------------------------------------------------
-- Table `employee_nurse`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `employee_nurse` ;


SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `employee_nurse` (
  `emp_number` INT NULL DEFAULT NULL,
  `nurse_code` INT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;


-- -----------------------------------------------------
-- Table `employee_physician`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `employee_physician` ;
```

```
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `employee_physician` (
  `emp_number` INT NULL DEFAULT NULL,
  `physician_code` INT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `illness`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `illness` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `illness` (
  `illness_code` INT NOT NULL,
  `illness_name` VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (`illness_code`))
```

```
ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;


-- -----------------------------------------------------

-- Table `medication`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `medication` ;


SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `medication` (

  `medication_code` INT NOT NULL,

  `medication_name` VARCHAR(255) NULL DEFAULT NULL,

  `unit_cost` FLOAT NULL DEFAULT NULL,

  `quantity_on_hand` INT NULL DEFAULT NULL,

  `quantity_on_order` INT NULL DEFAULT NULL,

  `year_to_date_usage` INT NULL DEFAULT NULL,

  PRIMARY KEY (`medication_code`))
```

```
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;


-- -----------------------------------------------------
-- Table `medication_priscription`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `medication_priscription` ;


SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `medication_priscription` (
  `medication_code` INT NULL DEFAULT NULL,
  `prescreption_code` INT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;
```

```
-- -----------------------------------------------------
-- Table `need_surgery_skill`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `need_surgery_skill` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `need_surgery_skill` (
  `surgery_code` INT NULL DEFAULT NULL,
  `skill_code` INT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `nurse`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `nurse` ;
```

```
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `nurse` (
  `nurse_code` INT NOT NULL,
  `grade` CHAR(1) NULL DEFAULT NULL,
  `years_of_experience` INT NULL DEFAULT NULL,
  PRIMARY KEY (`nurse_code`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `nurse_surgery`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `nurse_surgery` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `nurse_surgery` (
```

```
  `surgery_code` INT NULL DEFAULT NULL,
  `nurse_code` INT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;


-- -----------------------------------------------------
-- Table `owner`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `owner` ;


SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `owner` (
  `owner_code` INT NOT NULL,
  `owner_name` VARCHAR(255) NULL DEFAULT NULL,
  `owner_type` VARCHAR(50) NULL DEFAULT NULL,
  `owner_percentage` FLOAT NULL DEFAULT NULL,
  `headquarter` VARCHAR(255) NULL DEFAULT NULL,
```

```
  PRIMARY KEY (`owner_code`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;


-- -----------------------------------------------------
-- Table `patient`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `patient` ;


SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `patient` (
  `patient_number` INT NOT NULL AUTO_INCREMENT,
  `patient_name` VARCHAR(255) NULL DEFAULT NULL,
  `DOB` DATE NULL DEFAULT NULL,
  `gender` VARCHAR(10) NULL DEFAULT NULL,
  `ssn` CHAR(11) NULL DEFAULT NULL,
  `blood_type` VARCHAR(10) NULL DEFAULT NULL,
```

```
  `surgery_required` CHAR(1) NULL DEFAULT NULL,
  `stay_required` CHAR(1) NULL DEFAULT NULL,
  `address_line_1` VARCHAR(255) NULL DEFAULT NULL,
  `address_line_2` VARCHAR(255) NULL DEFAULT NULL,
  `city` VARCHAR(50) NULL DEFAULT NULL,
  `state` VARCHAR(50) NULL DEFAULT NULL,
  `zipcode` VARCHAR(15) NULL DEFAULT NULL,
  `country` VARCHAR(50) NULL DEFAULT NULL,
  PRIMARY KEY (`patient_number`))
ENGINE = InnoDB
AUTO_INCREMENT = 16
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;


-- -----------------------------------------------------
-- Table `patient_appointment`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `patient_appointment` ;
```

```
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `patient_appointment` (
  `patient_number` INT NULL DEFAULT NULL,
  `appointment_code` INT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `patient_illness`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `patient_illness` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `patient_illness` (
  `patient_number` INT NULL DEFAULT NULL,
  `illness_code` INT NULL DEFAULT NULL)
```

```
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;


-- -----------------------------------------------------
-- Table `patient_nurse`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `patient_nurse` ;


SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `patient_nurse` (
  `patient_number` INT NULL DEFAULT NULL,
  `nurse_code` INT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;
```

```
-- -----------------------------------------------------
-- Table `patient_physician`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `patient_physician` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `patient_physician` (
  `patient_number` INT NULL DEFAULT NULL,
  `physician_code` INT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `patient_priscription`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `patient_priscription` ;
```

```
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `patient_priscription` (
  `patient_number` INT NULL DEFAULT NULL,
  `prescription_code` INT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `patient_report`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `patient_report` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `patient_report` (
  `patient_number` INT NULL DEFAULT NULL,
  `report_code` INT NULL DEFAULT NULL)
```

```
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;


-- -----------------------------------------------------
-- Table `patient_room`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `patient_room` ;


SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `patient_room` (
  `room_code` INT NULL DEFAULT NULL,
  `patient_number` INT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;
```

```
-- -----------------------------------------------------
-- Table `patient_surgery`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `patient_surgery` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `patient_surgery` (
  `patient_number` INT NULL DEFAULT NULL,
  `surgery_code` INT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `physician`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `physician` ;
```

```
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `physician` (
  `physician_code` INT NOT NULL,
  `speciality` VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (`physician_code`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `physician_prescription`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `physician_prescription` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `physician_prescription` (
  `physician_code` INT NULL DEFAULT NULL,
```

```
  `prescription_code` INT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;


-- -----------------------------------------------------
-- Table `priscription`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `priscription` ;


SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `priscription` (
  `priscription_code` INT NOT NULL,
  `severity` CHAR(1) NULL DEFAULT NULL,
  PRIMARY KEY (`priscription_code`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

```
SHOW WARNINGS;


-- -----------------------------------------------------
-- Table `report`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `report` ;


SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `report` (
  `report_code` INT NOT NULL,
  `allergy_name` VARCHAR(255) NULL DEFAULT NULL,
  `bood_suger` VARCHAR(50) NULL DEFAULT NULL,
  `HDL` VARCHAR(50) NULL DEFAULT NULL,
  `LDL` VARCHAR(50) NULL DEFAULT NULL,
  `triglycerides` VARCHAR(50) NULL DEFAULT NULL,
  `risk` CHAR(1) NULL DEFAULT NULL,
  PRIMARY KEY (`report_code`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
```

```
COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;


-- -----------------------------------------------------
-- Table `room`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `room` ;


SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `room` (
  `room_code` INT NOT NULL,
  `wing` VARCHAR(5) NULL DEFAULT NULL,
  `nursing_unit` INT NULL DEFAULT NULL,
  `bed` CHAR(1) NULL DEFAULT NULL,
  PRIMARY KEY (`room_code`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

```
SHOW WARNINGS;


-- -----------------------------------------------------
-- Table `surgeon`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `surgeon` ;


SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `surgeon` (
  `surgeon_code` INT NOT NULL,
  `number_of_surgeries` INT NULL DEFAULT NULL,
  `speciality` VARCHAR(255) NULL DEFAULT NULL,
  `contract_start_date` DATE NULL DEFAULT NULL,
  `contract_end_date` DATE NULL DEFAULT NULL,
  PRIMARY KEY (`surgeon_code`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;
```

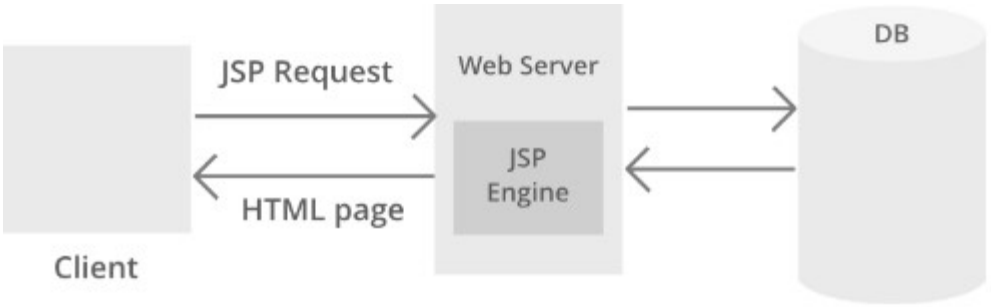
```
-- -----------------------------------------------------
-- Table `surgeon_surgery`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `surgeon_surgery` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `surgeon_surgery` (
  `surgeon_code` INT NULL DEFAULT NULL,
  `surgery_code` INT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `surgery`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `surgery` ;
```

```
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `surgery` (
  `surgery_code` INT NOT NULL,
  `surgery_name` VARCHAR(255) NULL DEFAULT NULL,
  `surgery_type` VARCHAR(50) NULL DEFAULT NULL,
  `surgery_time` DATETIME NULL DEFAULT NULL,
  `anatomical_location` VARCHAR(255) NULL DEFAULT NULL,
  `special_need` VARCHAR(255) NULL DEFAULT NULL,
  `category` VARCHAR(50) NULL DEFAULT NULL,
  `operation_theater_number` VARCHAR(10) NULL DEFAULT NULL,
  PRIMARY KEY (`surgery_code`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

SHOW WARNINGS;

-- -----------------------------------------------------
-- Table `surgery_skills`
```

```
-- -----------------------------------------------------

DROP TABLE IF EXISTS `surgery_skills` ;


SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `surgery_skills` (

  `skill_code` INT NOT NULL,

  `skill_name` VARCHAR(255) NULL DEFAULT NULL,

  `skill_description` VARCHAR(255) NULL DEFAULT NULL,

  PRIMARY KEY (`skill_code`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4_0900_ai_ci;


SHOW WARNINGS;


SET SQL_MODE=@OLD_SQL_MODE;

SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;

SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

## 4.2 Software Application Architecture, Design & Development

### Application Architecture:




## Application Design :

### Database Layer

The database layer stores all data used by the application. ER Diagram, Relational Schema Design and Relational Schema mention previously are used to create database layer.

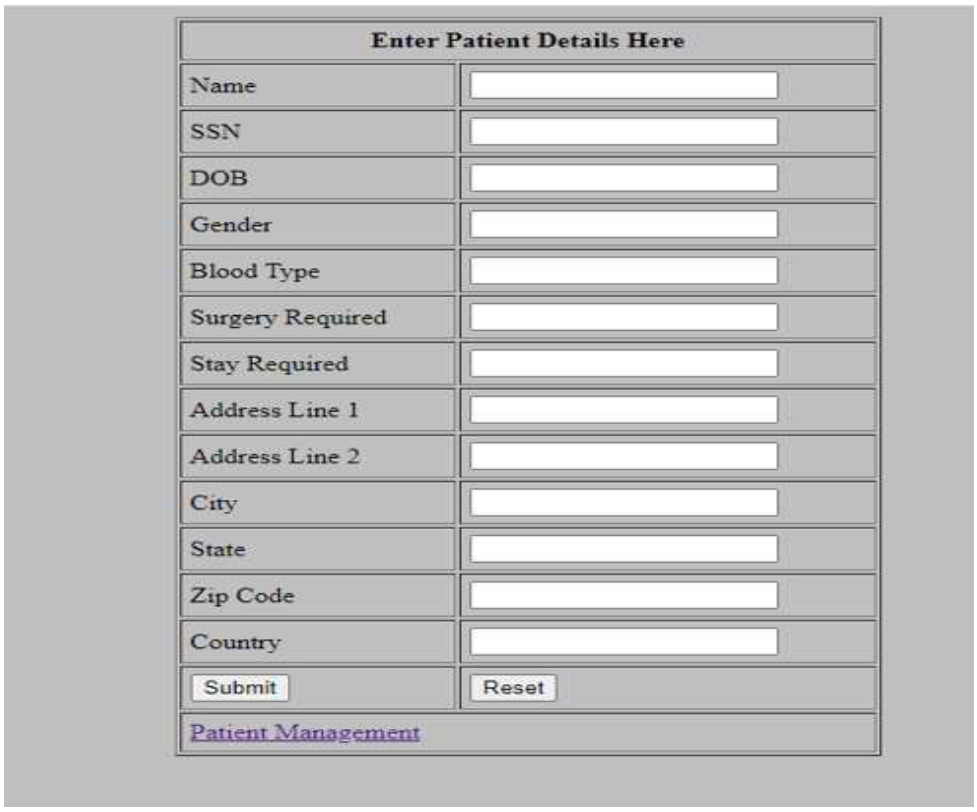
### Front and Middle Layer

As mentioned before, Frontend and Middle Layer are implemented using Core Java, JSP Servlets, JDBC and HTML.

## 4.3 Application Design Manual

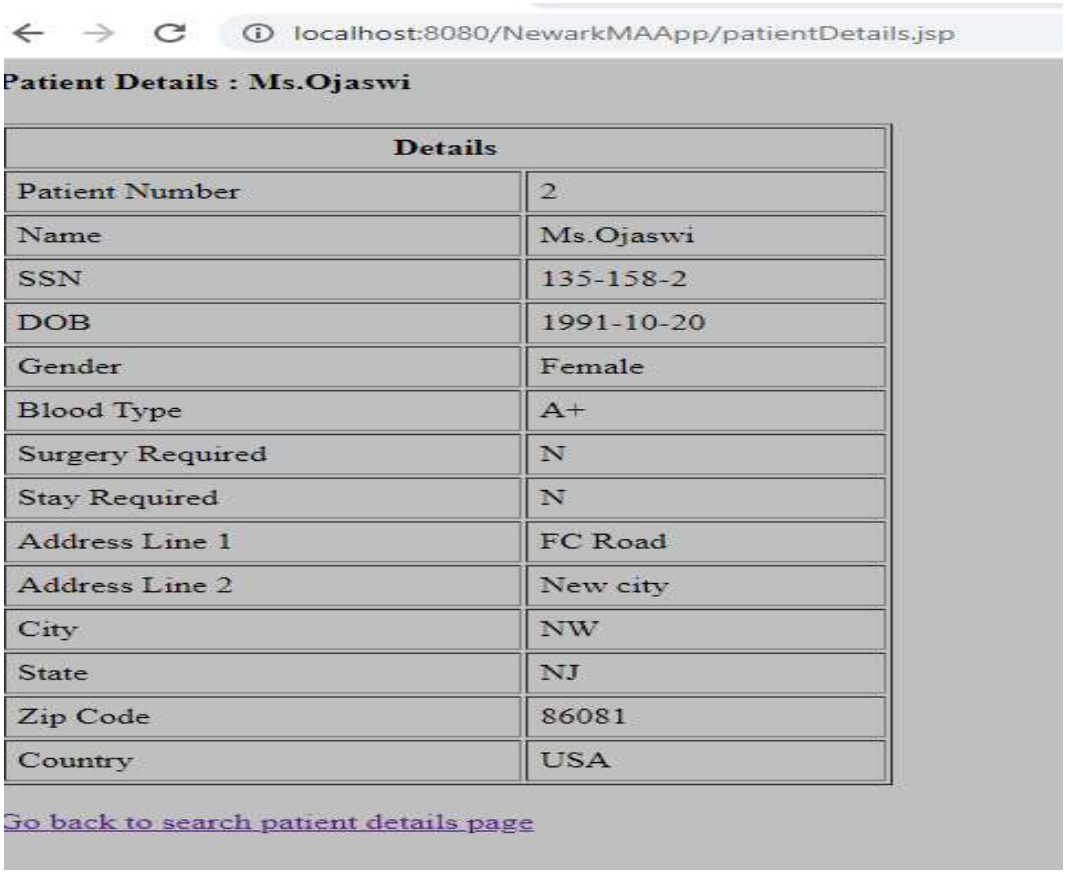
### Insert new patient:

To insert new patient, first need to click on insert a new patient, then it would show you an online registration form where you need to put patient details to register a new patient. Below is a frontend application design to register a new patient.



## Patient data retrieval:

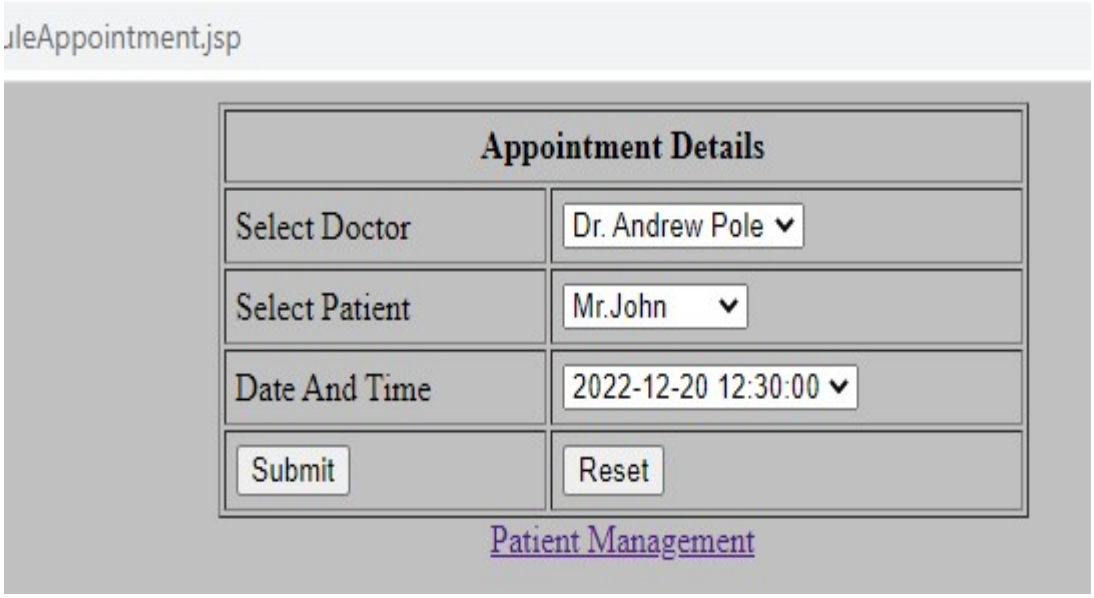
Hospital Management System application makes it possible to access all the data related to a patient via a system by the means of a few simple clicks. Information like patient history and if patient need surgery and stay required for that purpose, can be made visible to the user.




Above is a frontend application design to view patient details, after selecting patient management application, click on view patient information then add patient name to get required patient details;then the window shows patient number, name, SSN, DOB, Gender, Blood-type, Surgery required, Stay required, and Postal address.

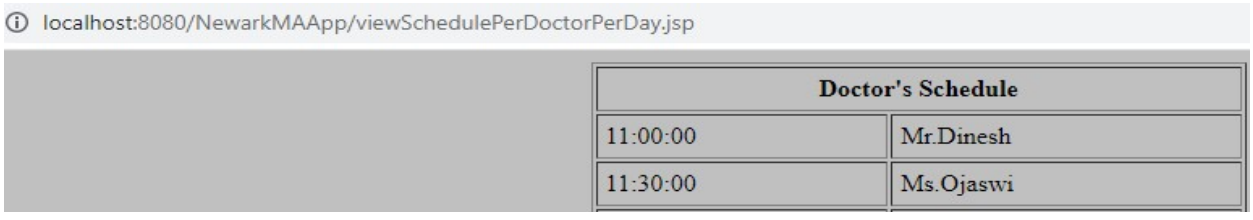
**Schedule an appointment with Doctor:**

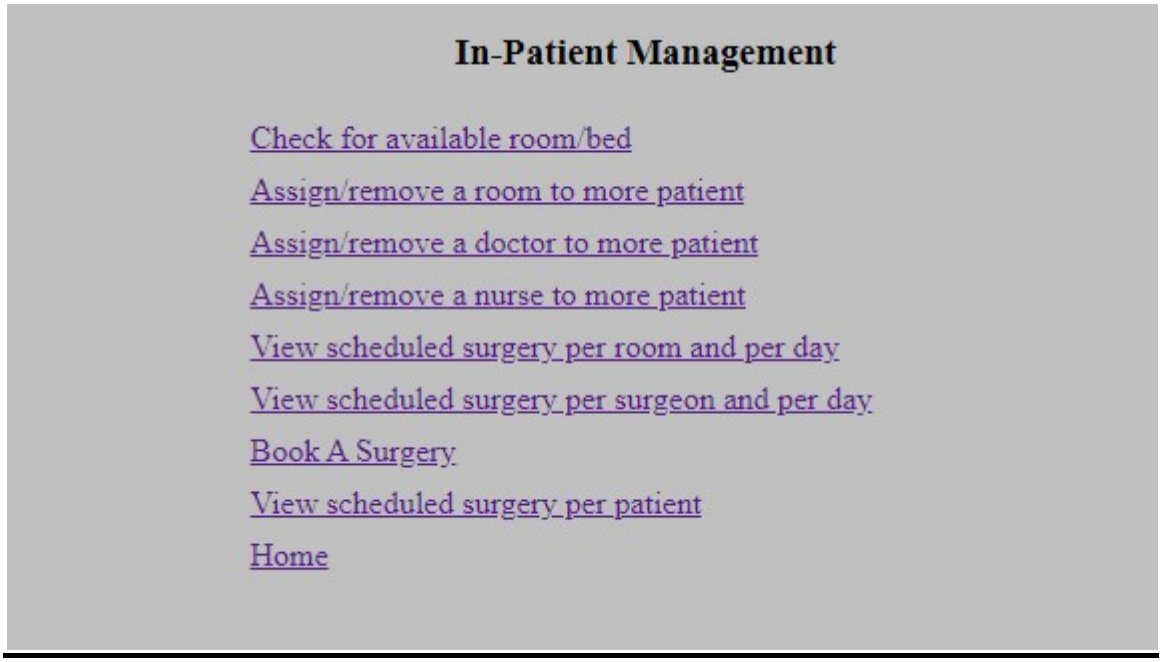
Once you clicked on Schedule an appointment with Doctor, it shows a window where it needs to select Doctor, patient, and date-time. Then submit it to schedule an appointment. Below is the frontend application design to schedule an appointment with doctor.



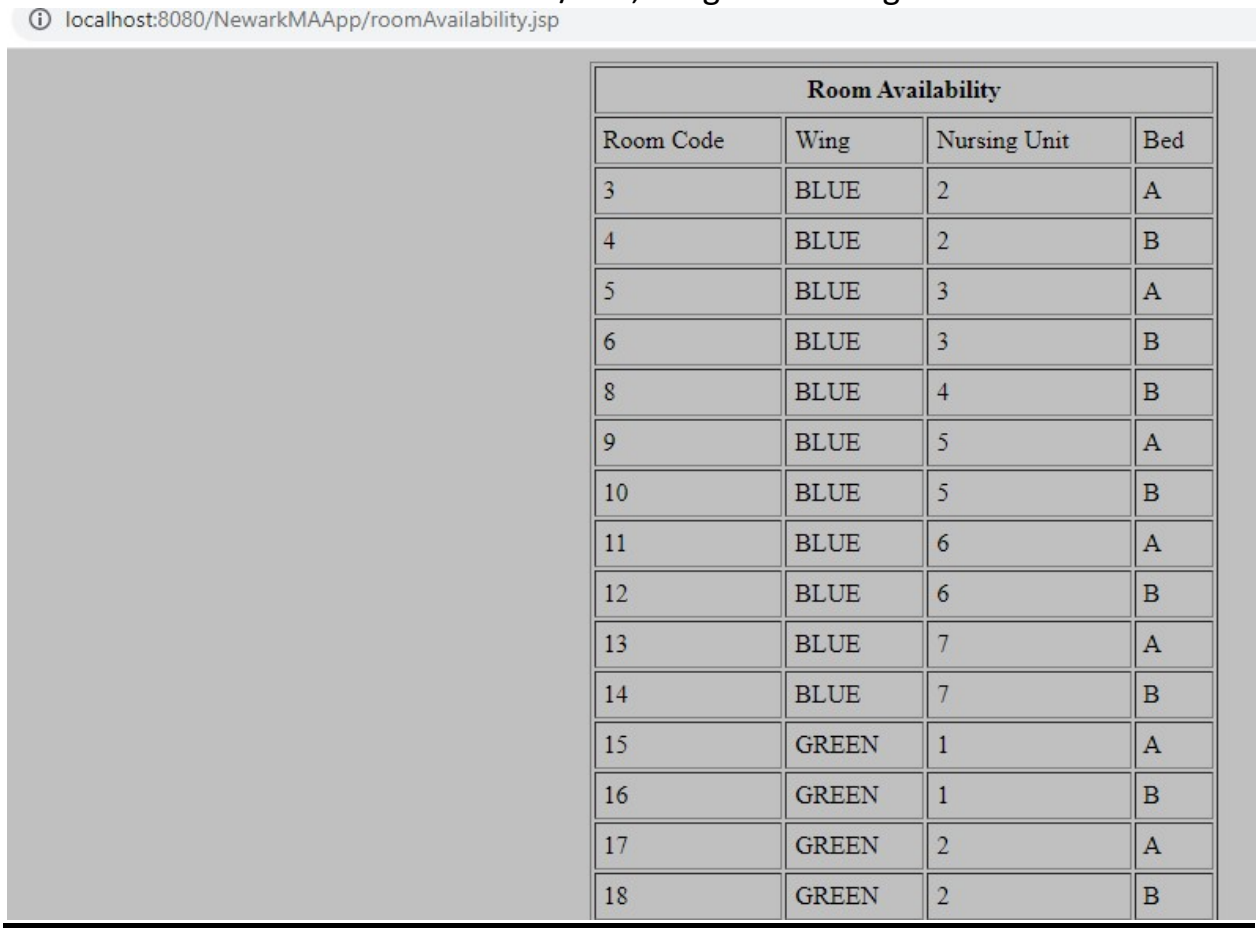

**View each Doctor schedule per day:**

In order to view doctor's schedule, one need to click on View scheduled per doctor per day and then select doctor from drop down list; it will get back a window with required information. The frontend design is,





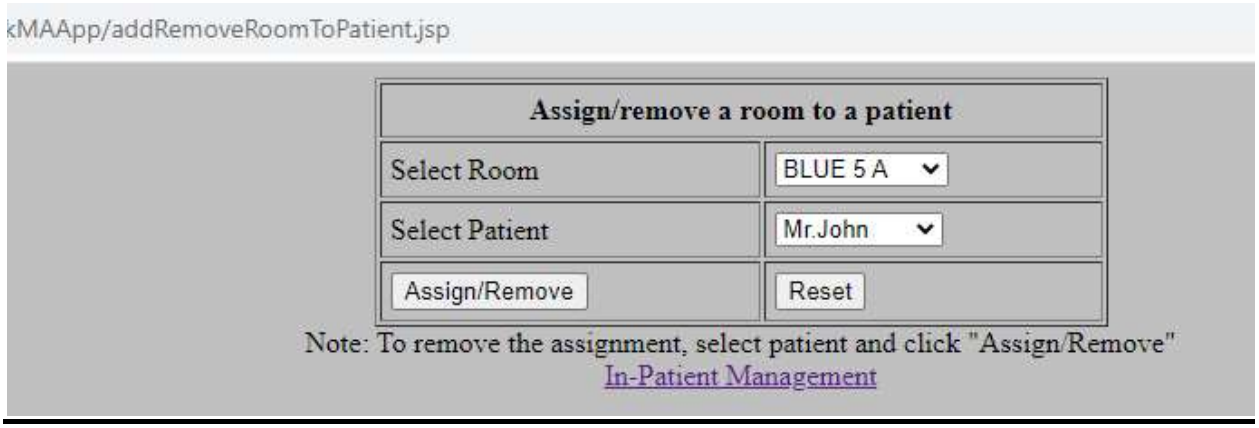

**Check for available room/bed:** To check room availability we just need to click on link check for available room/bed, we get following window




| Room Availability | | | |
|---|---|---|---|
| Room Code | Wing | Nursing Unit | Bed |
| 3 | BLUE | 2 | A |
| 4 | BLUE | 2 | B |
| 5 | BLUE | 3 | A |
| 6 | BLUE | 3 | B |
| 8 | BLUE | 4 | B |
| 9 | BLUE | 5 | A |
| 10 | BLUE | 5 | B |
| 11 | BLUE | 6 | A |
| 12 | BLUE | 6 | B |
| 13 | BLUE | 7 | A |
| 14 | BLUE | 7 | B |
| 15 | GREEN | 1 | A |
| 16 | GREEN | 1 | B |
| 17 | GREEN | 2 | A |
| 18 | GREEN | 2 | B |

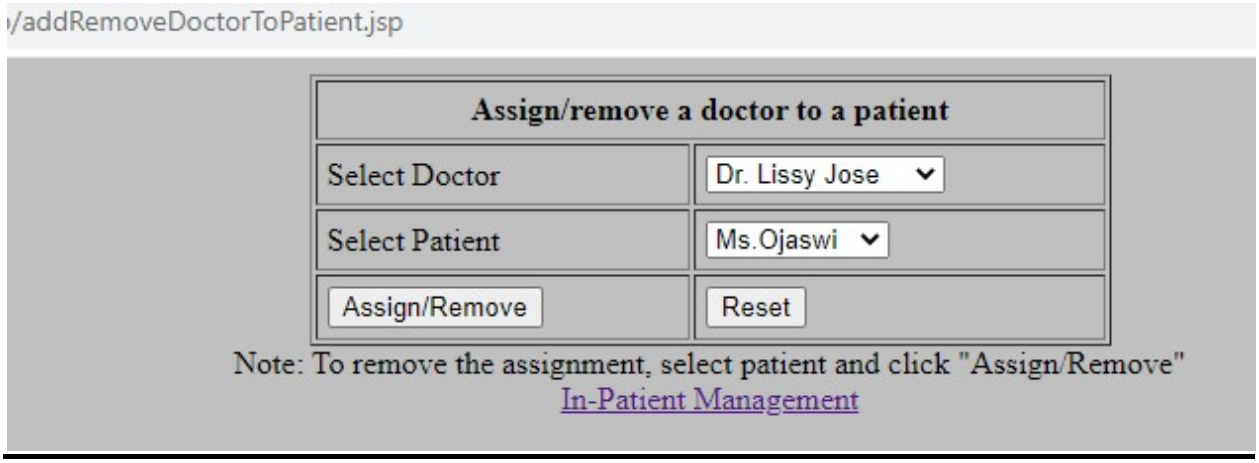
## Assign/Remove a room to more patients:

If staff wants to assign a room to a patient then he needs to click on Assign/remove a room to a patient then select room and patient from drop down list, then click on assign. And if staff wants to remove a patient then he needs to select patient name which we need to remove from drop down list and click on remove.



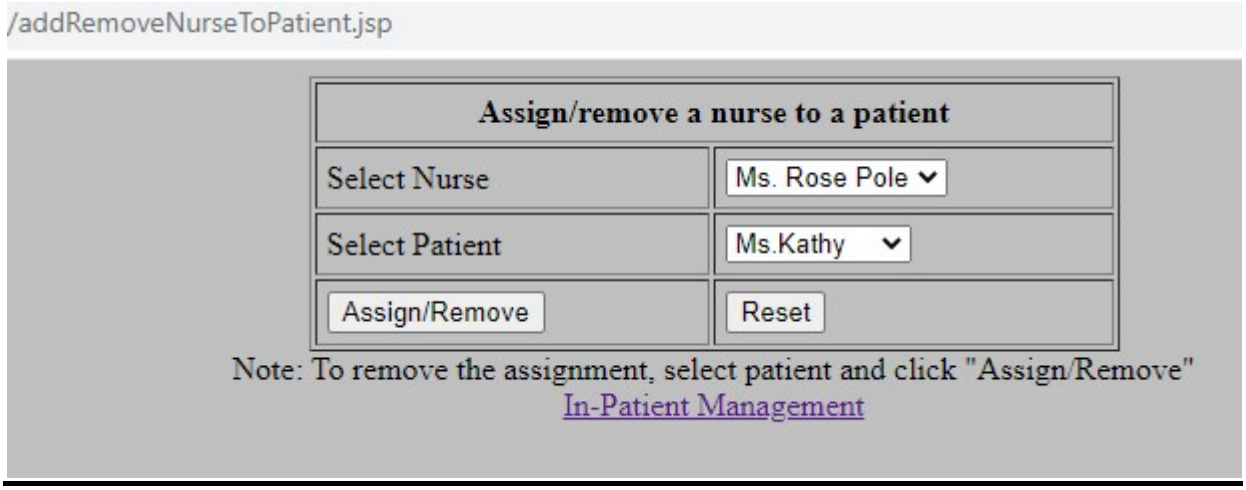

## Assign/Remove a doctor to more patients:

If staff wants to assign a doctor to a patient then he needs to click on Assign/remove a doctor to a patient then select doctor and patient from drop down list, then click on assign.



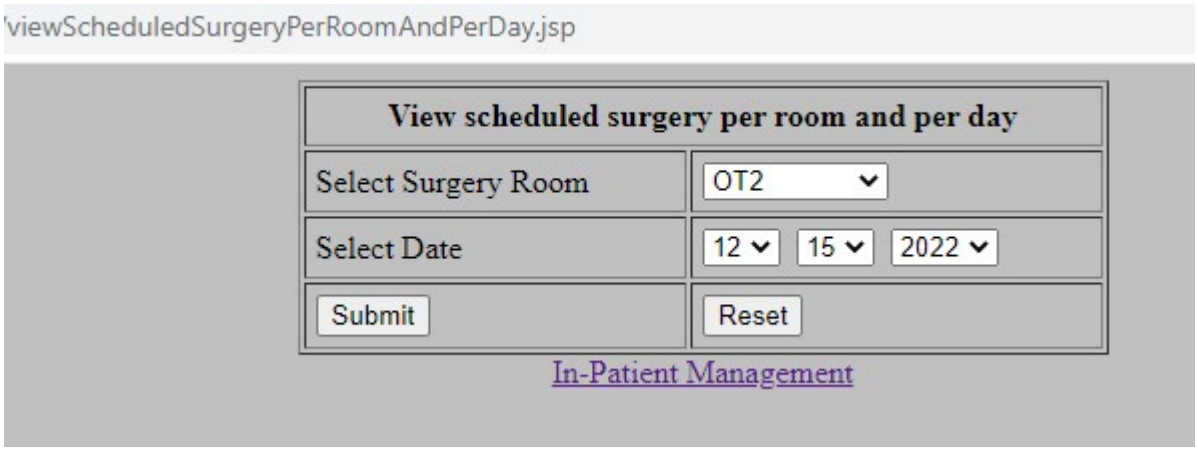
## Assign/Remove a nurse to more patients:

If staff wants to assign a nurse to a patient then he needs to click on Assign/remove a nurse to a patient then select nurse and patient from drop down list, then click on assign.



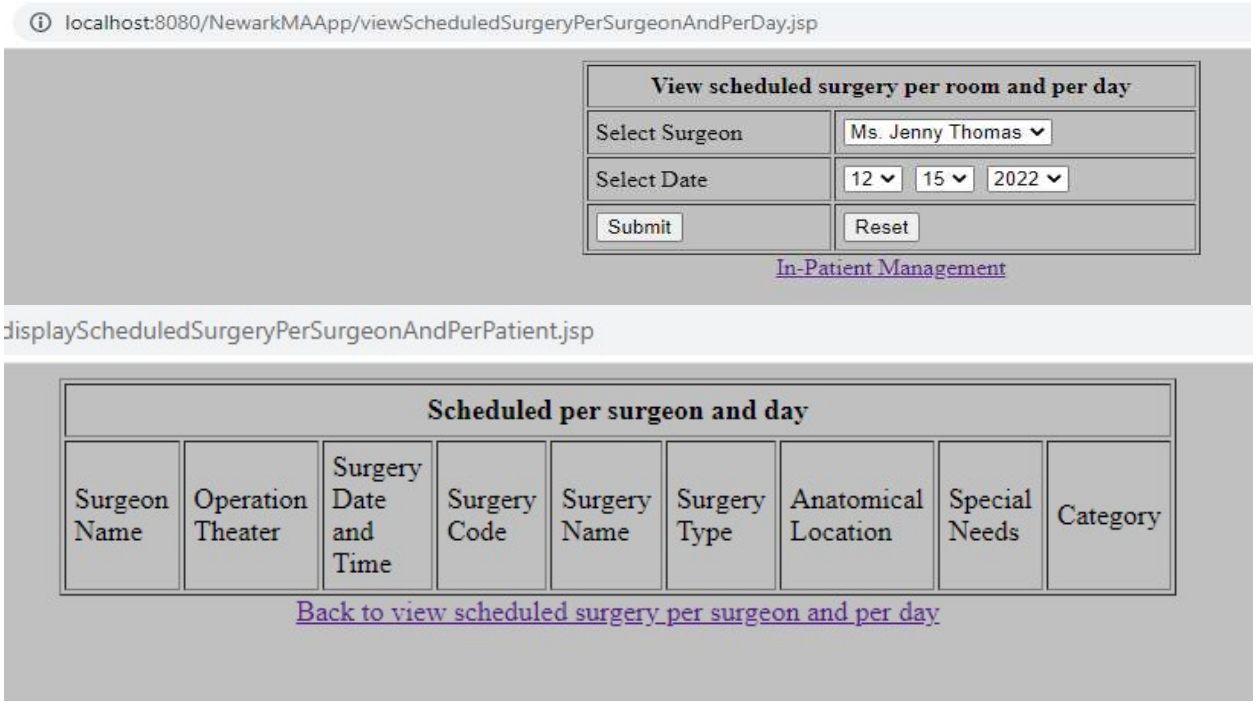

## View Scheduled Surgery per room per day:

Staff can check surgery room availability status with this link



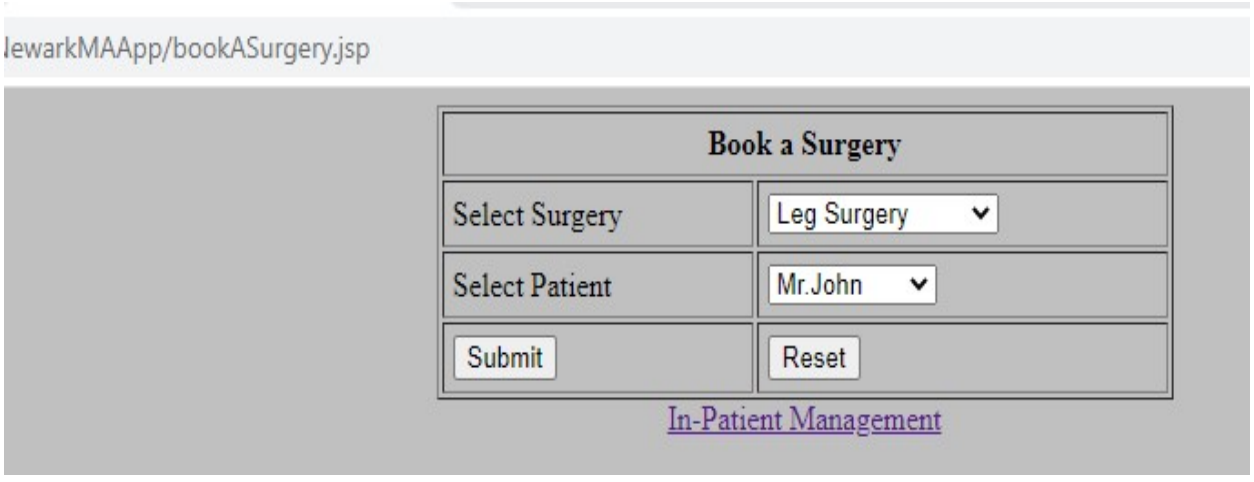
## View Scheduled Surgery per surgeon per day:

Staff can view scheduled surgery to check availability slot with this link




## Book a Surgery:

Staff can book a surgery by selecting surgery type and patient from drop down list



## View scheduled surgery per patient:

Staff can check scheduled surgery of specific patient by selecting patient name from drop down list when needed. Here we want to fetch Mr. Dinesh's scheduled surgeries, so we selected Dinesh as patient name from drop down list and we got back a window of Dinesh's scheduled surgeries as shown below.

viewScheduledSurgeryPerPatient.jsp

| View scheduled surgery per patient | |
|---|---|
| Select Patient | Mr.Dinesh |
| Submit | Reset |

In-Patient Management

/displayScheduledSurgeryPerPatient.jsp

| Staff Member By Job Type | | | | | | | |
|---|---|---|---|---|---|---|---|
| Surgery Code | Surgery Name | Surgery Type | surgery_time | Anatomical Location | Special Needs | Category | Operation Theater |
| 1 | General Surgery | General | 2022-12-01 14:00:00 | Foot | Full time Nurse | H | OT1 |
| 3 | Stomach Surgery | Stomach | 2022-12-01 10:00:00 | Stomach | 24 hours fasting | O | OT1 |

Back to view scheduled surgery per patient

/medicalStaffManagement.jsp

**Medical Staff Management**

Add Staff Member

Delete Staff Member

Schedule job shift

View stuff member per job type

Home

## Add Staff Management:

We can add new employee details when we assign new staff.

/addStaff.jsp

| Enter Staff Details Here | |
|---|---|
| Name | Mr. Allen |
| Type | SUPPORT_STAFF |
| SSN | |
| Gender | |
| Salary | |
| Address Line 1 | |
| Address Line 2 | |
| City | |
| State | |
| Zip Code | |
| Country | |
| Submit | Reset |

Medical Staff management

## Delete Staff Member:

Staff can delete any employee details when he/she resign.

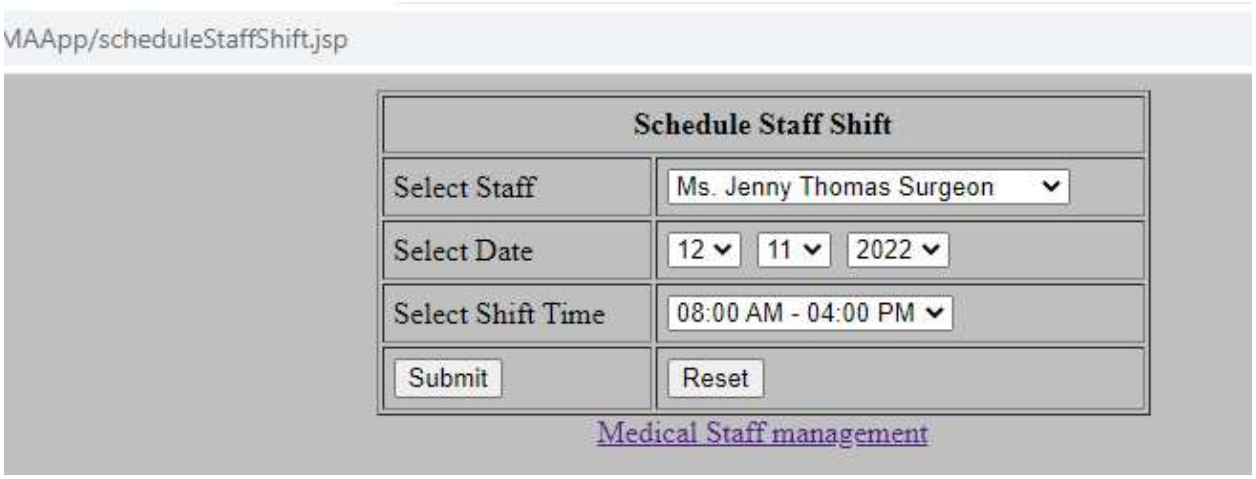
MAApp/deleteStaff.jsp

| Delete Staff | |
|---|---|
| Select Staff | Mr. Lauren Graham Staff-Nurse |
| Delete | Reset |

Medical Staff management

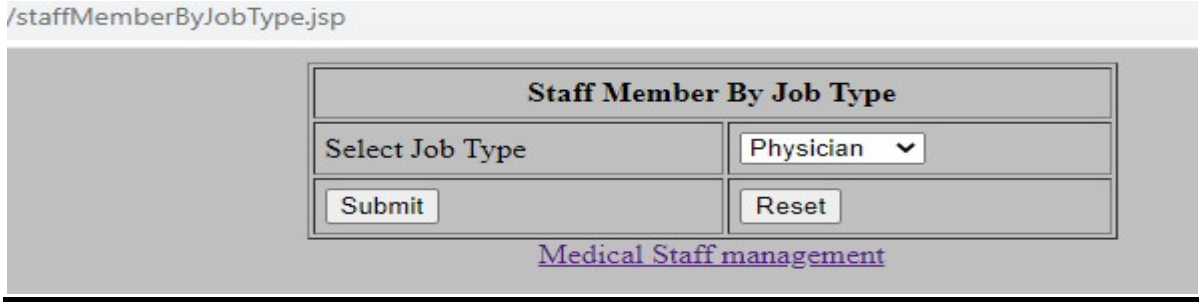
## Schedule job shift:

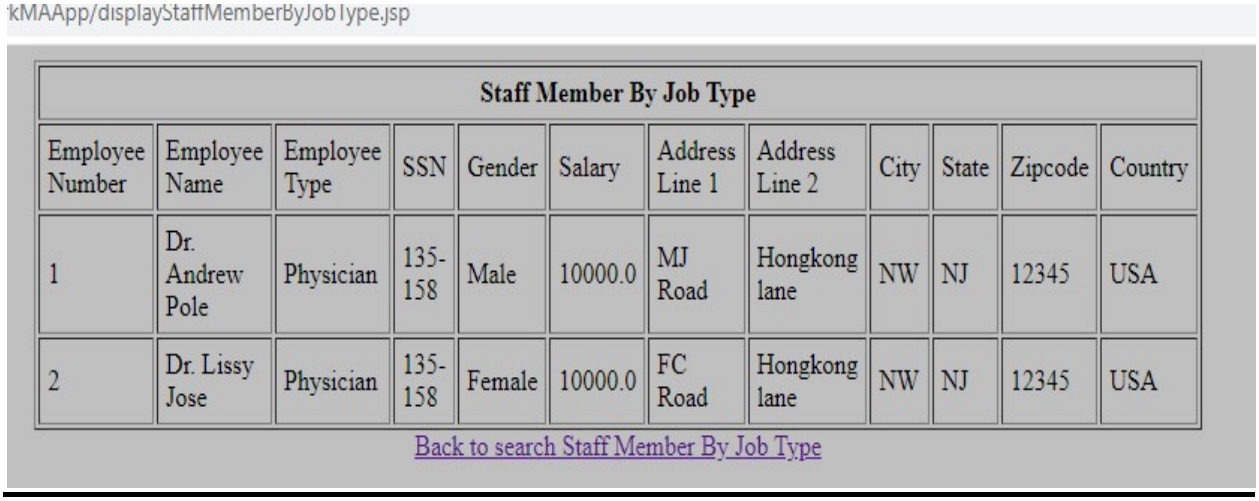
Employee can allot proper job shift to respective employee by selecting staff, date, and shift time from drop down list




## View staff member per job type:

Employee can view staff members per job type like here below, we select physician from drop down list, so we got each physician details.



## 5. Appendix:

| TABLE | ATTRIBUTE | DESCRIPTION |
|---|---|---|
| Employee | | Store information about hospital's employees |
| | emp_number | Employee number |
| | emp_name | Employee name |
| | emp_type | Employee type |
| | Ssn | Social Security Number of employee |
| | gender | Whether employee is male or female |
| | salary | Salary of employee |
| | | |
| Patient | | Stores patient's details |
| | patient_number | Patient number |
| | patient_name | Patient name |
| | DOB | Date of Birth |
| | gender | Whether Patient is Male or Female |
| | Ssn | Social Security Number of Patient |
| | blood_type | Blood group of Patient |

| | | |
|---|---|---|
| | surgery_required | Whether Patient require Surgery or not |
| | stay_required | Whether Patient need to stay or not |
| | | |
| appointment | | Appointment code with time and date specified for patient to schedule an appointment with Doctor |
| | appointment_code | Specified appointment code to schedule an appointment |
| | appointment_date | Appointment date specified |
| | appoinmtment_time | Appoinmtment time specified |
| | | |
| Nurse | | Store Nurse details |
| | nurse_code | Nurse code to distinguish from other nurse staff |
| | Grade | Nurse's Performance grade |
| | Years of experience | Nurse's professional experience |
| | | |
| Physician | | Store Physicians details |
| | physician_code | Physician's code to distinguish from other |
| | speciality | Physician's Specialization |

| | | |
|---|---|---|
| Surgeon | | Stores surgeon details |
| | surgeon_code | Surgeon's code to distinguish from other |
| | number_of_surgeries | Total numbers of surgeries carried |
| | speciality | Surgeon's speciality |
| | contract_start_date | As surgeon is on contract basis, it stores contract's start date |
| | contract_end_date | As surgeon is on contract basis, it stores contract's end date |
| | | |
| Surgery | | Each surgery details is stored |
| | surgery_code | Surgery code to distinguish from other |
| | surgery_name | Surgery name |
| | surgery_type | Store information on which body part the surgery is undertaken |
| | surgery_time | It stores the duration of surgery undertaken |
| | anatomical_location | Store information on which organ the anatomical operation is carried out |
| | special_need | Store information about what patient primarily need |
| | operation_theater_number | In which OT the operation is carried out |

## 6. Conclusion:

This Hospital Management System Application helps manage the information related to health care and aids in the job completion of health care providers effectively. This project will provide a convenient proposition and approach to a database for fetching required data. In spite of its many advantages, Hospital Management System Application can be improved in many ways according to requirement.