CS313 : DataBases and Information Systems Lab

Project 2

Sourabh Bhosale (200010004)
Dibyashu Kashyap (200010013)

October 30, 2022

1 Overview of the project

Hospital Management System:

This is a system that has the following functionalities:

- Add new doctor
- Add new patient
- Add new staff
- Book an appointment option for admin
- Book an appointment option for patient
- Updating details option for doctor
- View personal details option for doctor
- View personal details option for patient

This hospital management system can be used by people who needs treatment according to the allergies or disease they have. First they will have to register themselves as a patient through admin, where they have to give all the information about themselves like name, address, allergies, height, weight etc. Then appointment can be booked by admin or patient can do it by himself after logging in. We also have an option to add a doctor or hospital staff by taking respective information like name, department names they want to work in, speciality for doctor. Doctor can update his details later on if required. And doctor and patient can view the details of themselves after logging in.

2 Database schema, ER Model, Integrity Constraints and views, Users

2.1 Database schema and Integrity Constraints

Table	Primary Key	Domain of P.K.	Foreign keys (Referencing table)	Not Null
patient	patient_id	VARCHAR	-	-
address	patient_id	VARCHAR	patient_id references patient	-
allergies	patient_id	VARCHAR	patient_id references patient	-
dept	dept_name	VARCHAR	-	-
doctor	doctor_id	VARCHAR	dept_name references dept	-
staff	staff_id	VARCHAR	dept_name references dept	-
appointment	appointment_id	VARCHAR	patient_id references patient doctor_id references doctor	-
adminpass	admin_id	VARCHAR	-	-
docpass	doc_id pass	VARCHAR	doctor_id references doctor doctor_id references doctor	-
patpass	patient_id pass	VARCHAR	patient_id references patient patient_id references patient	-

Table 1: Table of constraints

2.2 Users

There are 3 types of users in our hospital management system:

- Admin : An admin after loging-in with the correct credentials can add a new doctor, new patient, new staff and an appointment for a patient.
- Doctor : A doctor after loging-in with the correct credentials can update his/her details and also view his/her details.
- Patient : A patient after loging-in with the correct credentials can add an appointment and also view the details.

2.3 ER Model

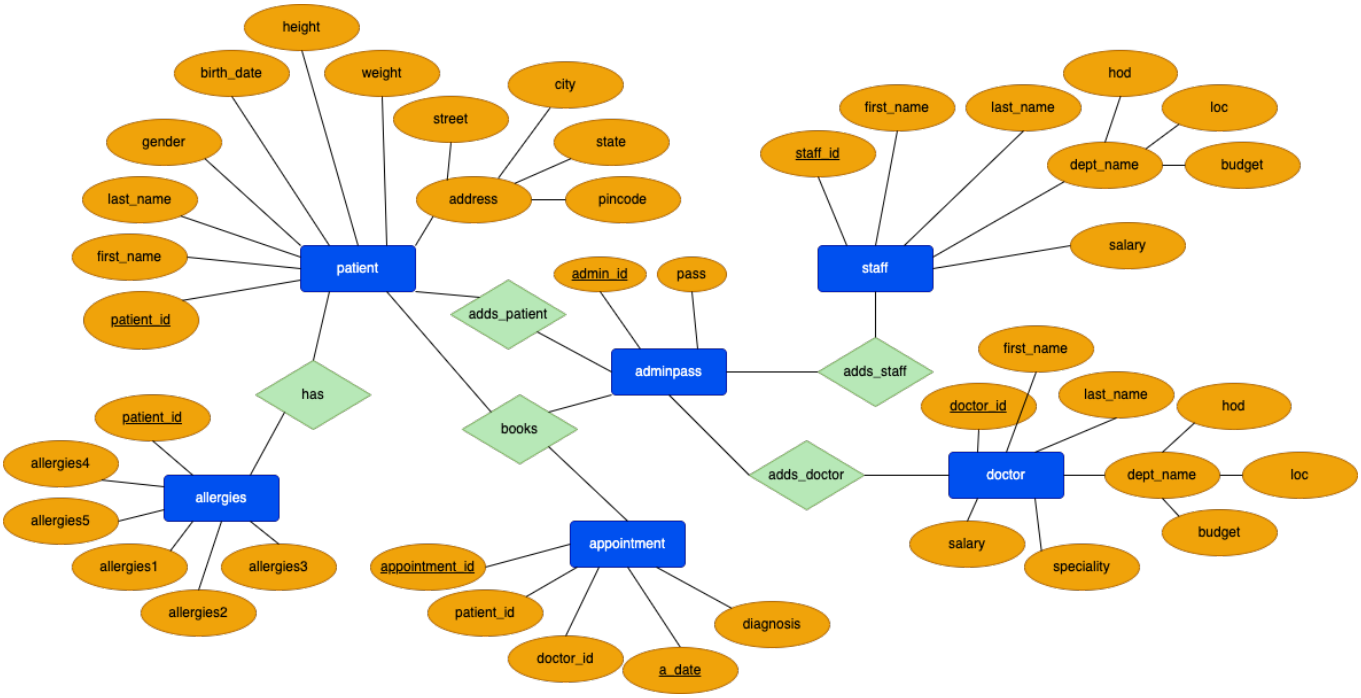

Figure 1: ER Diagram

3 Relational Database design(Tables creation)

```
mysql> DESCRIBE patient;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| patient_id | varchar(8)  | NO   | PRI | NULL    |       |
| first_name | varchar(50) | YES  |     | NULL    |       |
| last_name  | varchar(50) | YES  |     | NULL    |       |
| gender     | varchar(10) | YES  |     | NULL    |       |
| birth_date | date        | YES  |     | NULL    |       |
| height     | int         | YES  |     | NULL    |       |
| weight     | int         | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
7 rows in set (0.05 sec)
```

Figure 2: `patient` table

```
mysql> DESCRIBE address;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| patient_id | varchar(8)  | NO   | PRI | NULL    |       |
| street     | varchar(20) | YES  |     | NULL    |       |
| city       | varchar(20) | YES  |     | NULL    |       |
| state      | varchar(20) | YES  |     | NULL    |       |
| pincode    | varchar(10) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

Figure 3: `address` table

```
mysql> DESCRIBE allergies;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| patient_id | varchar(8)  | NO   | PRI | NULL    |       |
| allergies1 | varchar(20) | YES  |     | NULL    |       |
| allergies2 | varchar(20) | YES  |     | NULL    |       |
| allergies3 | varchar(20) | YES  |     | NULL    |       |
| allergies4 | varchar(20) | YES  |     | NULL    |       |
| allergies5 | varchar(20) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

Figure 4: `allergies` table

```
mysql> DESCRIBE doctor;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| doctor_id  | varchar(8)  | NO   | PRI | NULL    |       |
| first_name | varchar(50) | YES  |     | NULL    |       |
| last_name  | varchar(50) | YES  |     | NULL    |       |
| dept_name  | varchar(20) | YES  | MUL | NULL    |       |
| speciality | varchar(30) | YES  |     | NULL    |       |
| salary     | int         | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

Figure 5: `doctor` table

```
mysql> DESCRIBE staff;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| staff_id   | varchar(8)  | NO   | PRI | NULL    |       |
| first_name | varchar(50) | YES  |     | NULL    |       |
| last_name  | varchar(50) | YES  |     | NULL    |       |
| dept_name  | varchar(20) | YES  | MUL | NULL    |       |
| salary     | int         | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

Figure 6: `staff` table

```
mysql> DESCRIBE dept;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| dept_name | varchar(20) | NO   | PRI | NULL    |       |
| hod       | varchar(20) | YES  |     | NULL    |       |
| loc       | varchar(20) | YES  |     | NULL    |       |
| budget    | int         | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Figure 7: `dept` table

```
mysql> DESCRIBE appointment;
+----------------+------------+------+-----+---------+-------+
| Field          | Type       | Null | Key | Default | Extra |
+----------------+------------+------+-----+---------+-------+
| appointment_id | varchar(8)  | NO   | PRI | NULL    |       |
| patient_id     | varchar(8)  | YES  | MUL | NULL    |       |
| doctor_id      | varchar(8)  | YES  | MUL | NULL    |       |
| a_date         | varchar(10) | NO   |     | NULL    |       |
| diagnosis      | varchar(50) | YES  |     | NULL    |       |
+----------------+------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

Figure 8: `appointment` table

```
mysql> DESCRIBE adminpass;
+----------+------------+------+-----+---------+-------+
| Field    | Type       | Null | Key | Default | Extra |
+----------+------------+------+-----+---------+-------+
| admin_id | varchar(20) | NO   | PRI | NULL    |       |
| pass     | varchar(20) | YES  |     | NULL    |       |
+----------+------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Figure 9: `adminpass` table

```
mysql> DESCRIBE docpass;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| doc_id | varchar(8)  | NO   | PRI | NULL    |       |
| pass   | varchar(50) | NO   | PRI | NULL    |       |
+--------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Figure 10: `docpass` table

```
mysql> DESCRIBE patpass;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| pat_id | varchar(20) | NO   | PRI | NULL    |       |
| pass   | varchar(20) | NO   | PRI | NULL    |       |
+--------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Figure 11: `patpass` table

4 Languages and Technologies used

4.1 Technologies used

- Eclipse IDE
- Visual Studio Code
- Java JDBC
- J2EEE
- MySQL Database
- Servlets and JSP files.
- Apache Tomcat 8.5 Server

4.2 Languages used

- Java (for Servlets)
- HTML in JSP files
- CSS
- MySQL
- XML

4.3 Libraries used

- Bootstrap CSS Library
- MySQL Library (.jar)

5 Interface Designs

In this hospital management system, we have used vanilla css along with the bootstrap library, HTML fields and buttons for text fields and referencing different pages.

6 Workflow and Pages

- At start, we will be at Home page (`Home.jsp`) where different options like Admin login, Doctor Login, Patient Login will be there.
- First let's look into Admin login (`adminlog.jsp`), if we enter correct login credentials, then it will proceed further and give options like Add a doctor, Add a patient, Add a staff and Book an appointmnt using `AdminlogServlet` .
- But if the login credentials are incorrect, then it will show error message to enter the correct ones.
- Admin can add a doctor (`adddoc.jsp`), add a patient (`addpat.jsp`), add a staff (`addstaff.jsp`) and book an appointment (`addapp.jsp`). Each of the things will be controlled by servlet files like (`AdddocServlet.java`), (`AddpatServlet.java`), (`AddstaffServlet.java`) and (`AddappServlet.java`)
- For each of the things, primary key constraints should be followed and if not it will show error message.
- Now, let's have a look at Doctor login (`doclog.jsp`), if we enter correct login credentials, then it will proceed further and give options like update the details (`updatedoc.jsp`) and show the details (`showdoc.jsp`) using (`DoctorlogServlet`). For updating details, sure to follow primary key constraints and if not it will show error message.
- But if the login credentials are incorrect for doctor, then it will show error message to enter the correct ones.
- Now for the Patient login (`patlog.jsp`), if we enter correct login credentials, then it will proceed further and give options like book an appointment (`addapp.jsp` and show the details (`showpat.jsp` using (`PatientlogServlet.java`). For booking an appointment, make sure to follow primary key constraints and if not it will show error message.
- If the login credentials are incorrect for patient, then it will show error message to enter the correct ones.

7 Add the relevant Screenshots

```
mysql> CREATE DATABASE hospital_db;
Query OK, 1 row affected (0.03 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| Assign3_users      |
| hospital_db        |
| information_schema |
| Library            |
| mydb               |
| mysql              |
| performance_schema |
| sys                |
| university         |
+--------------------+
9 rows in set (0.01 sec)

mysql> USE hospital_db;
Database changed
mysql> SELECT DATABASE();
+-------------+
| DATABASE()  |
+-------------+
| hospital_db |
+-------------+
1 row in set (0.00 sec)
```

Figure 12: Creating new database `hospital_db`

```
mysql> SHOW TABLES;
+-----------------------+
| Tables_in_hospital_db |
+-----------------------+
| address               |
| adminpass             |
| allergies             |
| appointment           |
| dept                  |
| docpass               |
| doctor                |
| patient               |
| patpass               |
| staff                 |
+-----------------------+
10 rows in set (0.00 sec)
```

Figure 13: Tables inside `hospital_db`

```
mysql> SELECT * FROM adminpass;
+----------+----------+
| admin_id | pass     |
+----------+----------+
| 10101    | test     |
| 20202    | password |
| 30303    | testtest |
+----------+----------+
3 rows in set (0.00 sec)
```

Figure 14: Credentials present for Admin login in the database

```
mysql> SELECT * FROM dept;
+-----------+------------+------+--------+
| dept_name | hod        | loc  | budget |
+-----------+------------+------+--------+
| Brain     | Dr.Brain   | 1F   |  90000 |
| Heart     | Dr.Heart   | 2F   |  95000 |
| Kidney    | Dr.Kidney  | 3F   |  80000 |
| Liver     | Dr.Liver   | 1F   |  70000 |
| Lungs     | Dr.Lungs   | 2F   |  90000 |
| Stomach   | Dr.Stomach | 3F   |  85000 |
+-----------+------------+------+--------+
6 rows in set (0.00 sec)
```

Figure 15: Initially inserted department table values

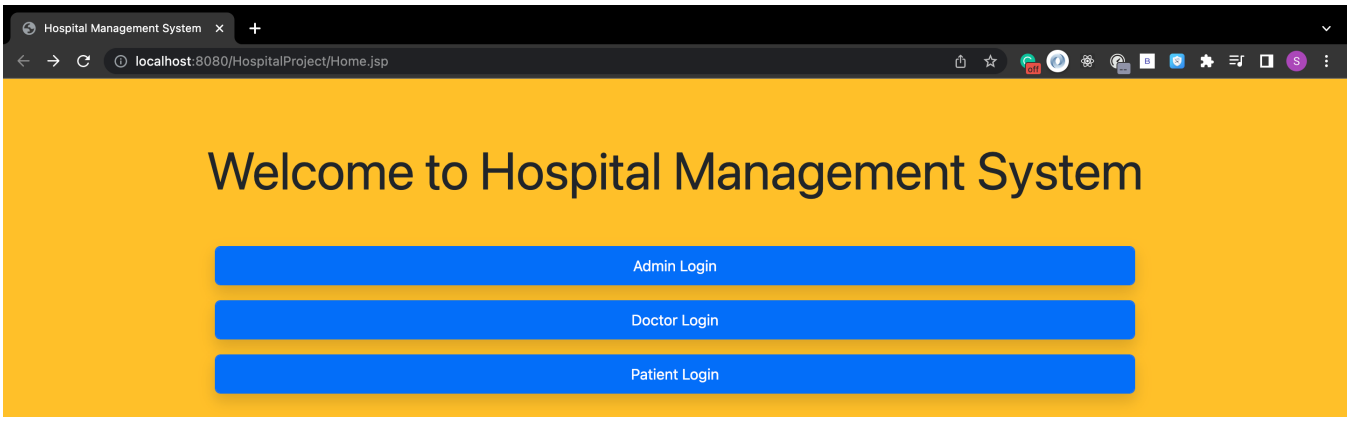

Figure 16: Home page

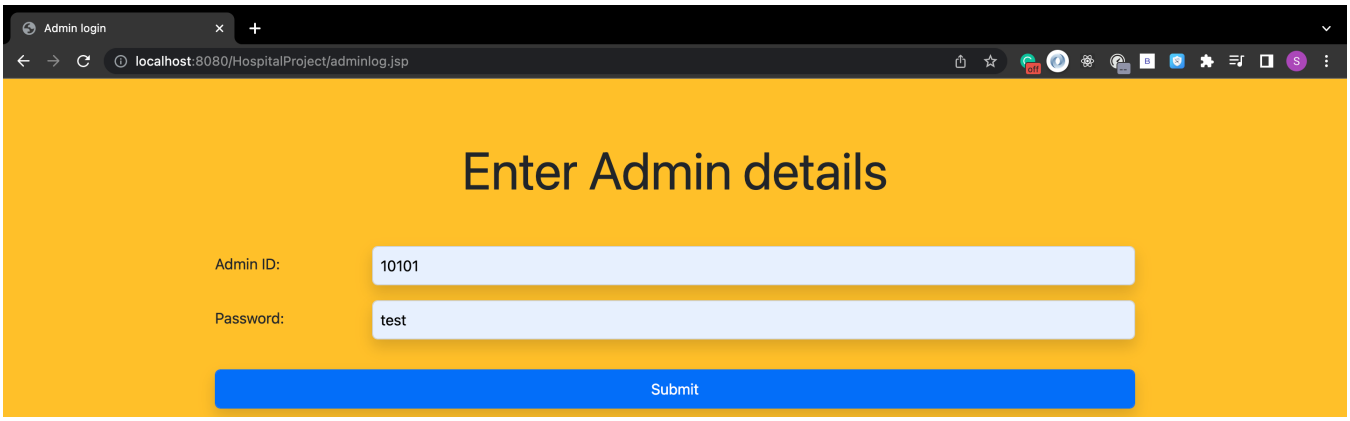

Figure 17: Admin login page(using correct credentials)

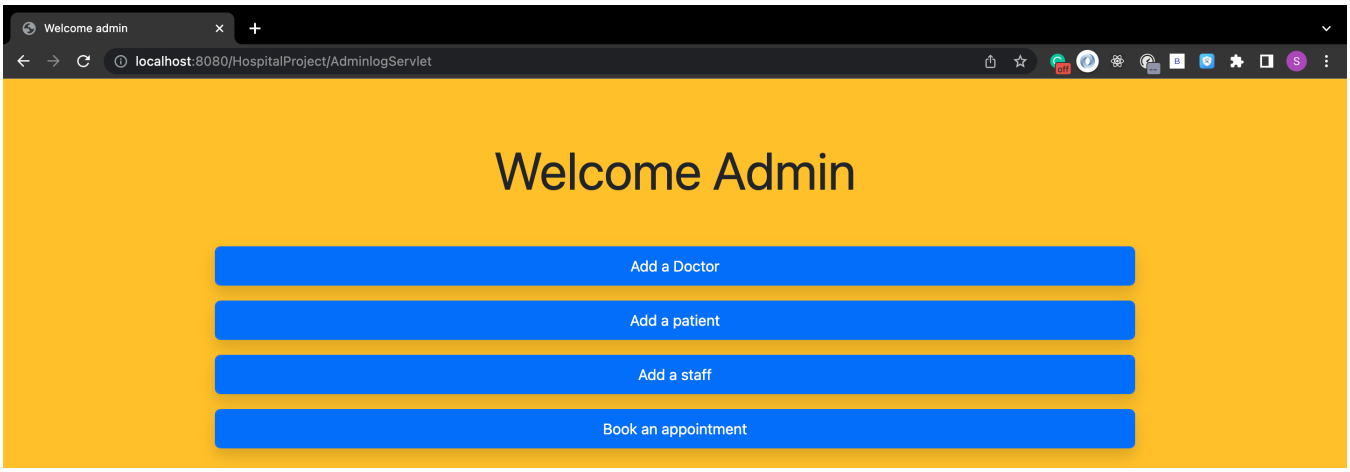

Figure 18: Page after successful admin login

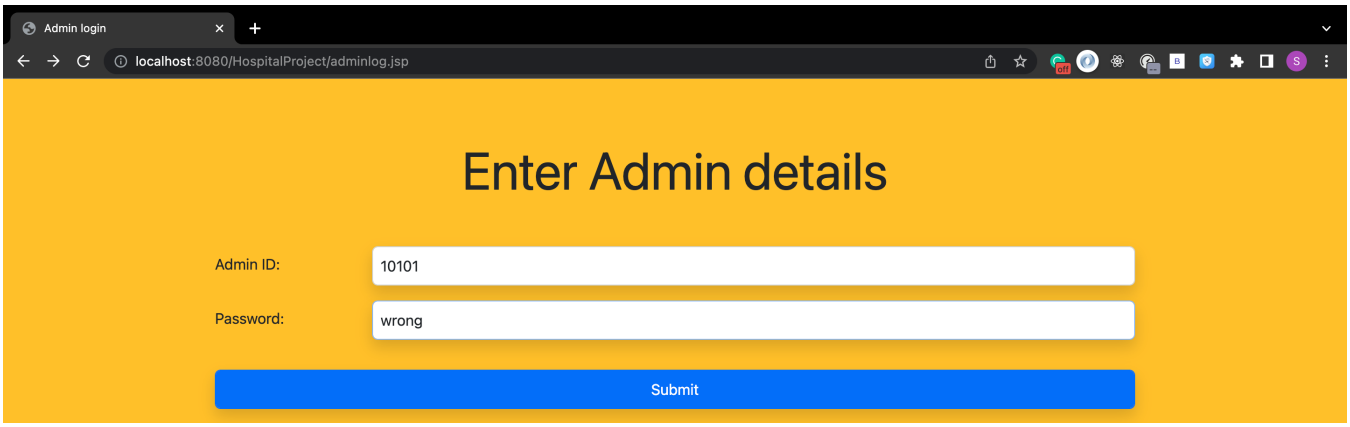

Figure 19: Admin login page(using incorrect credentials)

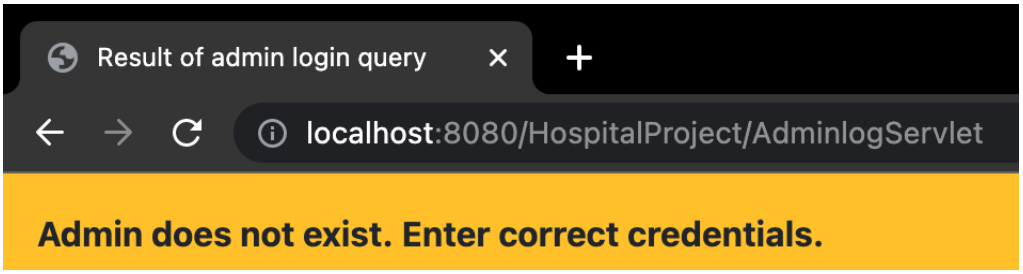

Figure 20: Result of incorrect login credentials

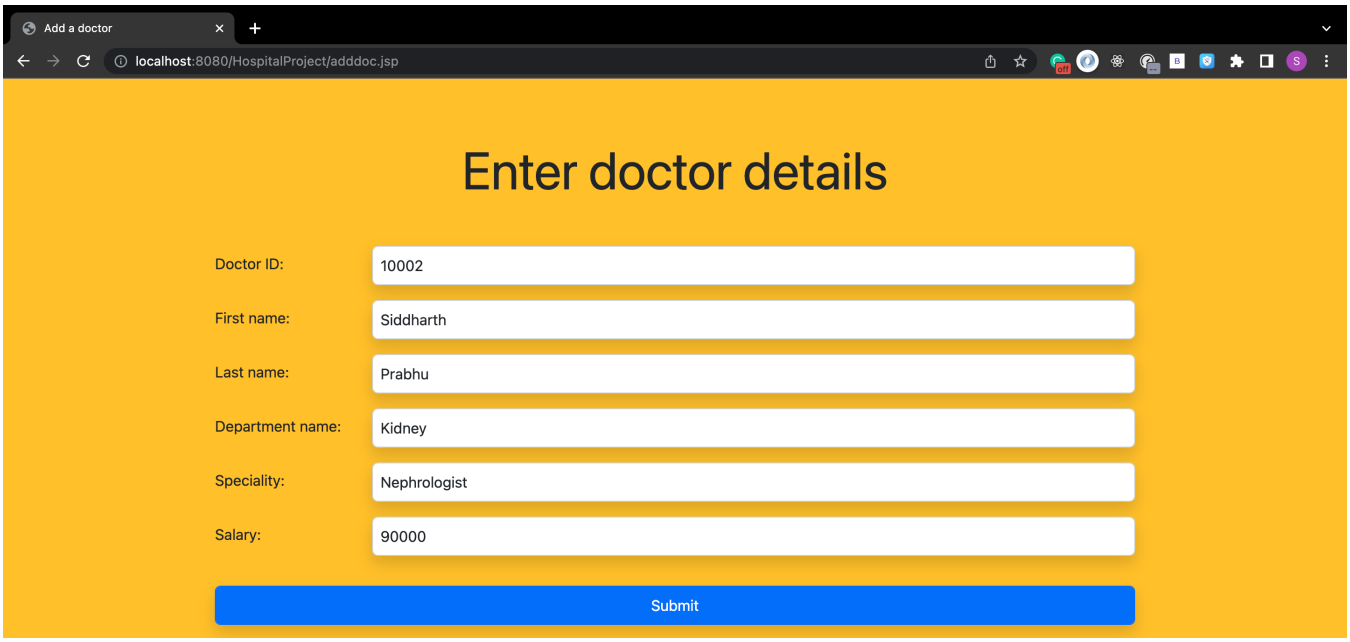

Figure 21: Adding a doctor

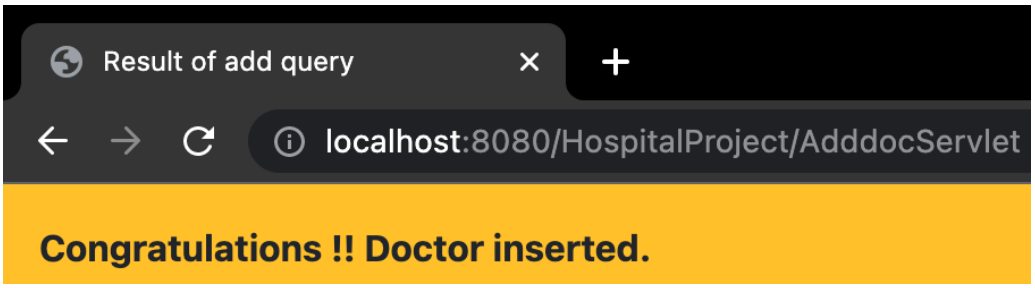

Figure 22: Successful insertion of doctor

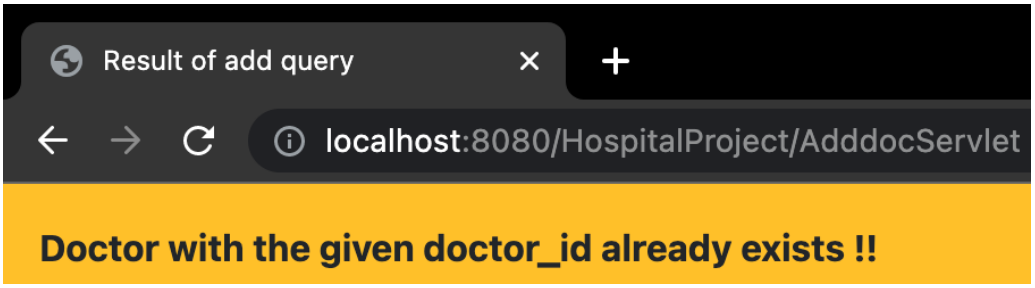

Figure 23: Error message if we try to violate the PK constraint

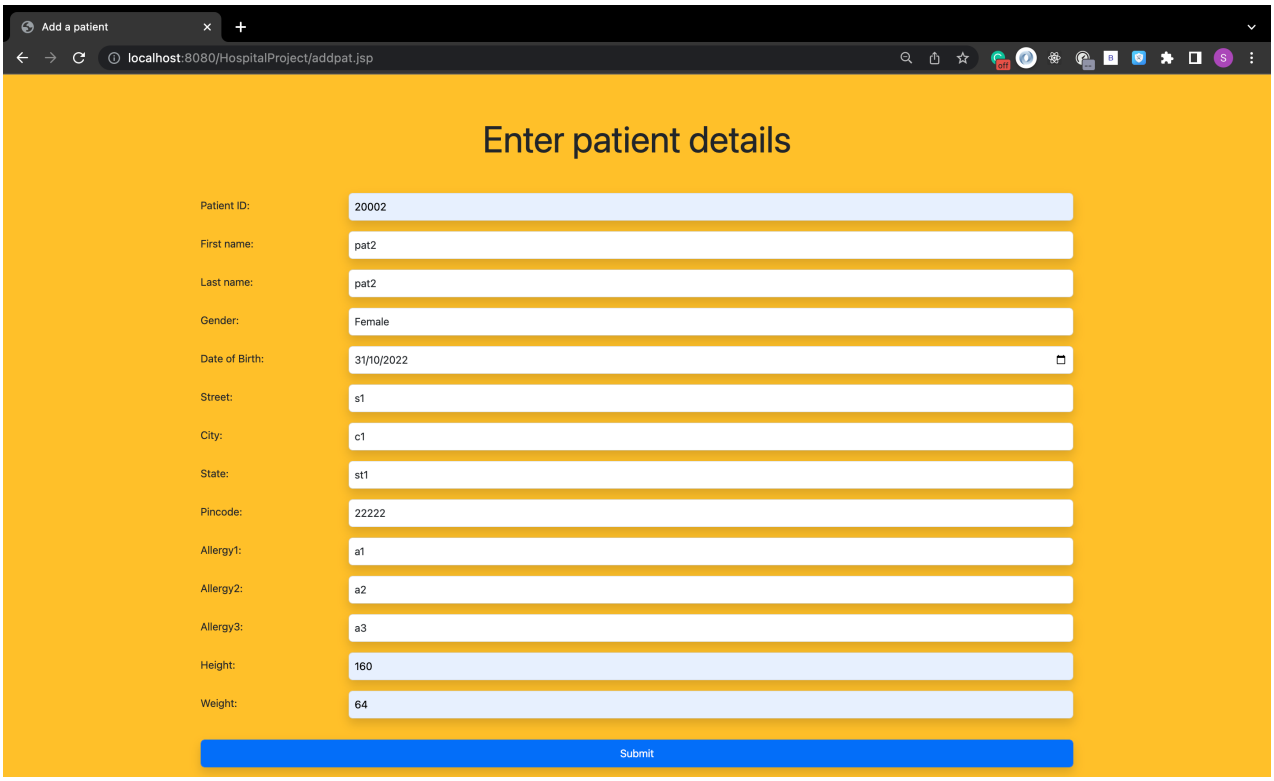

Figure 24: Adding a patient

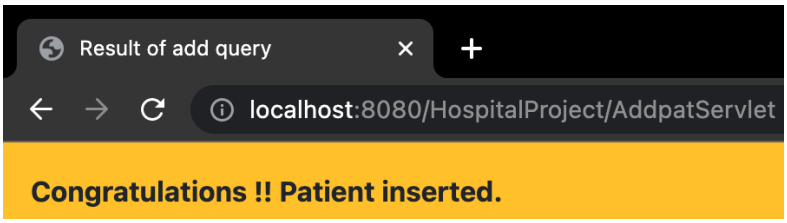

Figure 25: Successful insertion of patient

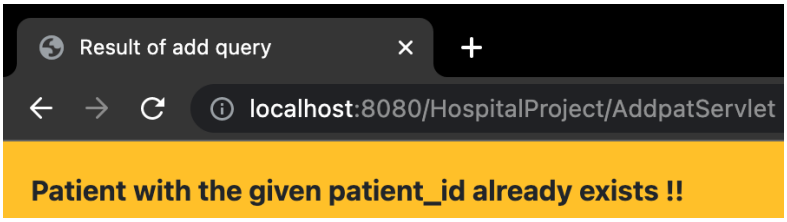

Figure 26: Error message if we try to violate the PK constraint

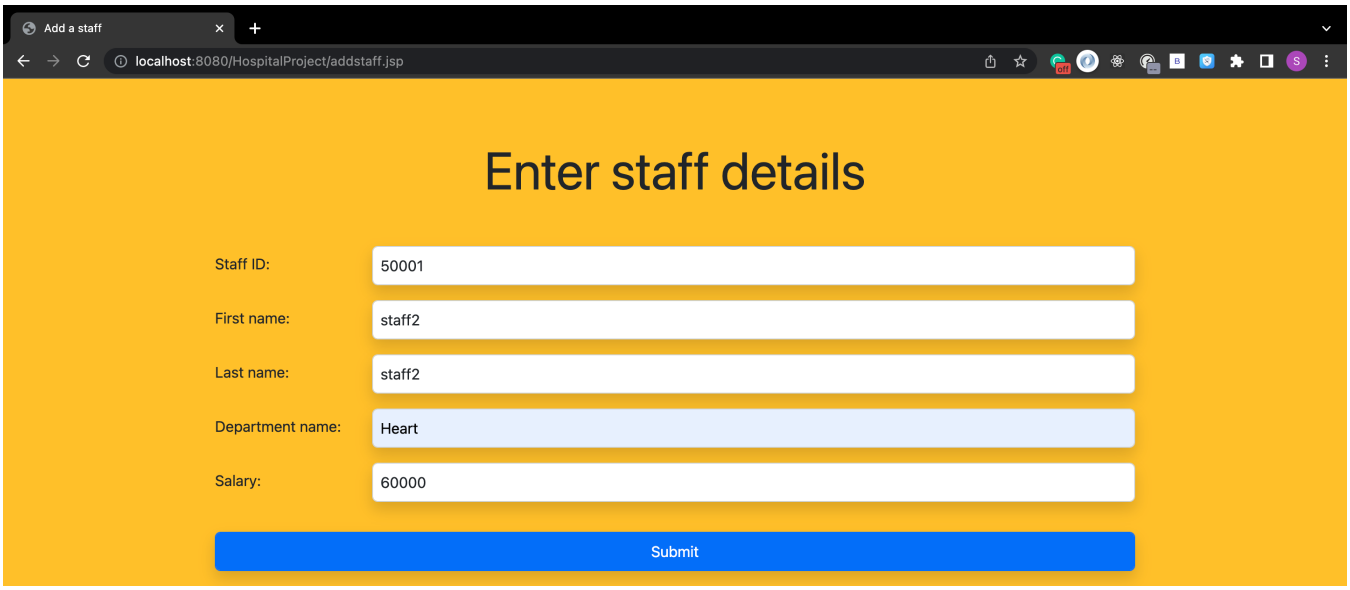

Figure 27: Adding a staff

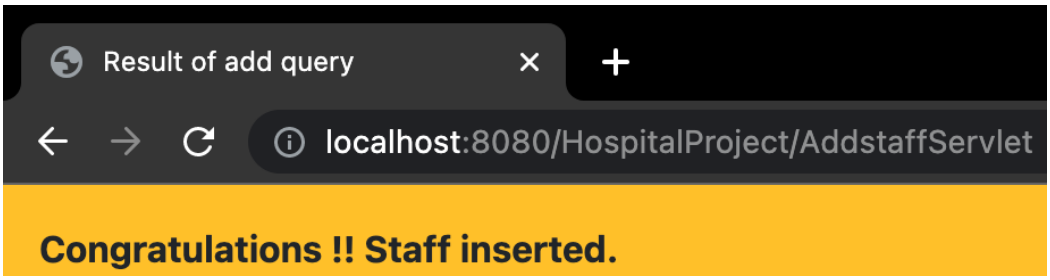

Figure 28: Successful insertion of staff

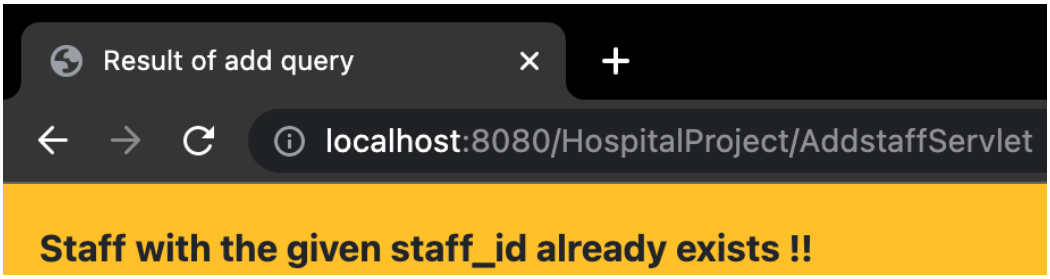

Figure 29: Error message if we try to violate the PK constraint

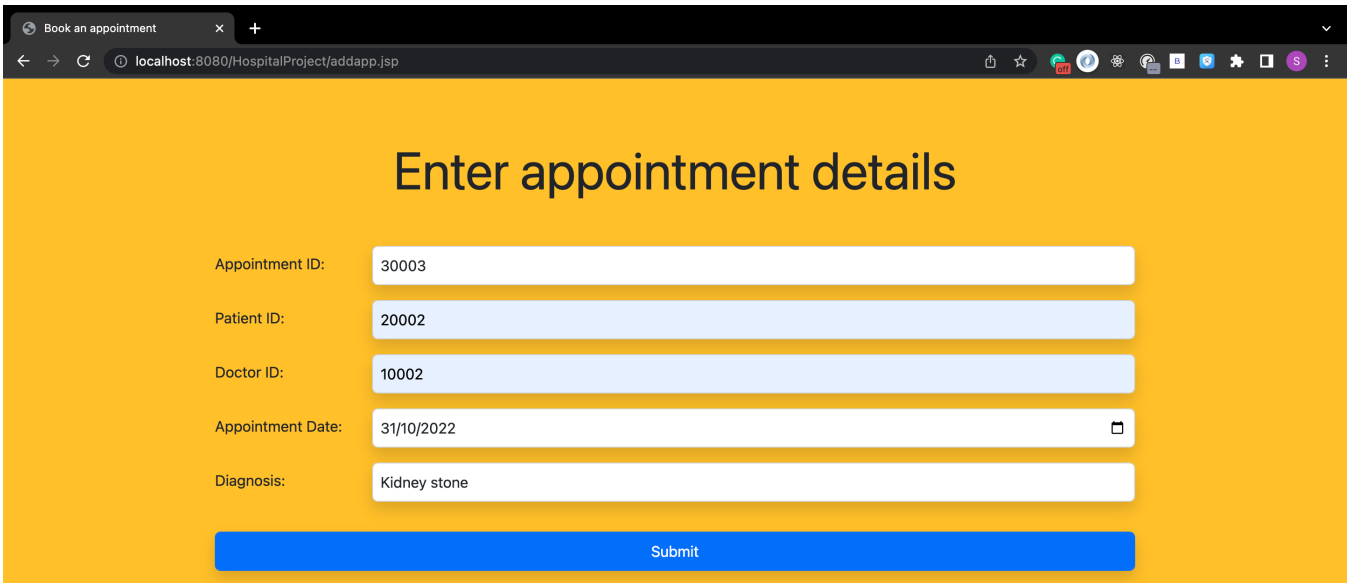

Figure 30: Booking an appointment (by admin)

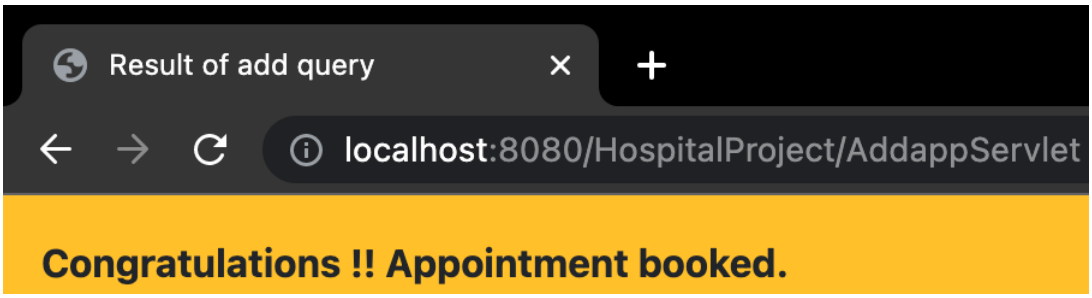

Figure 31: Successful booking

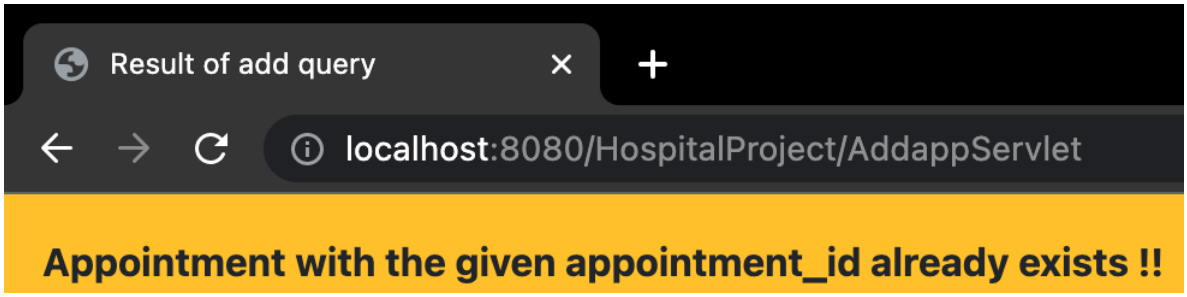

Figure 32: Error message if we try to violate the PK constraint

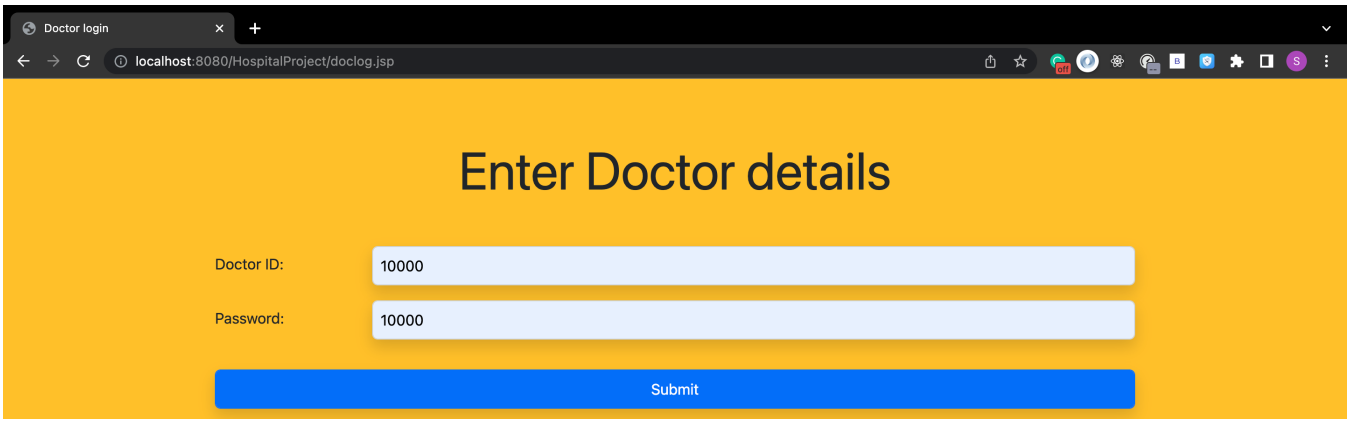

Figure 33: Doctor login page(using correct credentials)

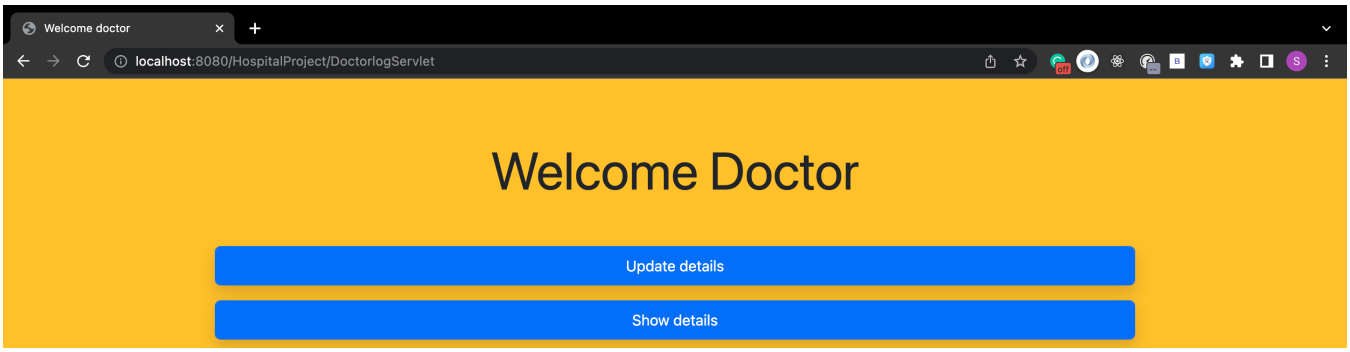

Figure 34: Page after successful doctor login

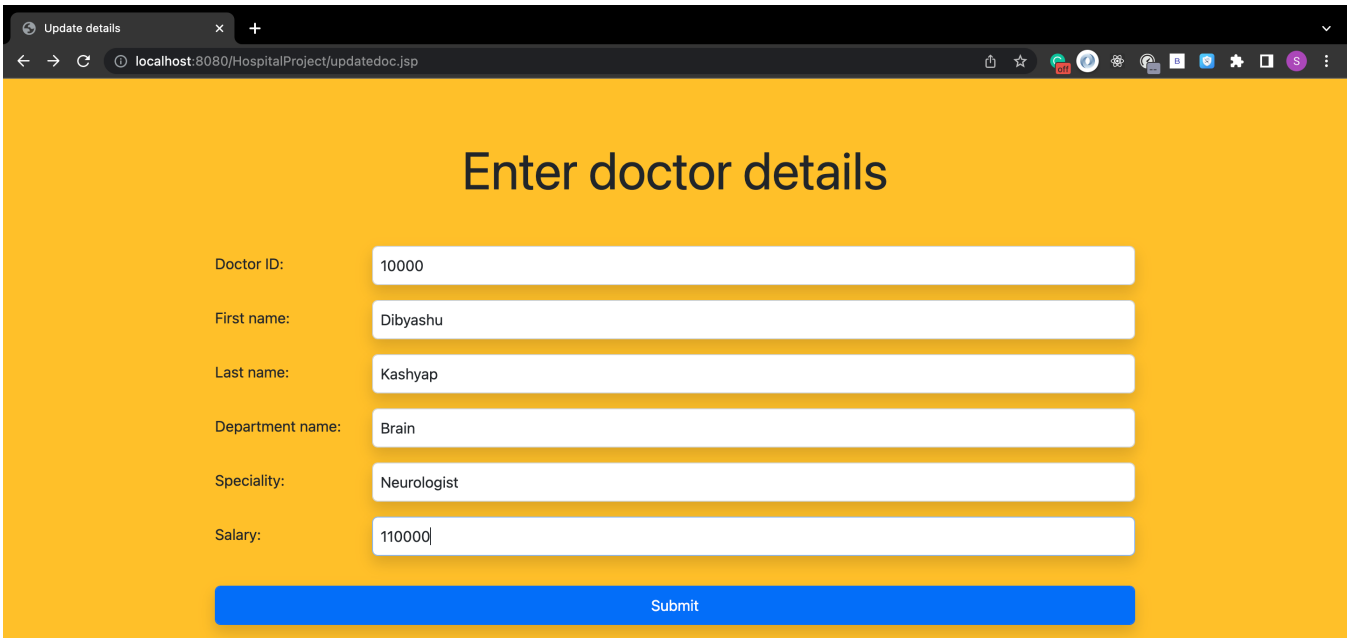

Figure 35: Updating doctor details

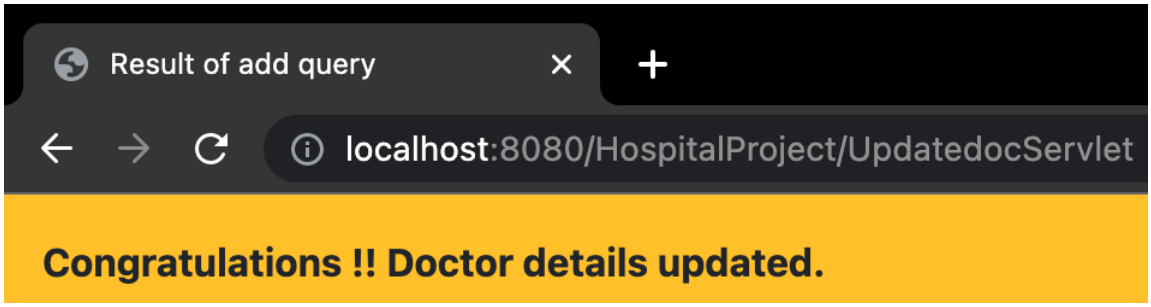

Figure 36: Result of update query

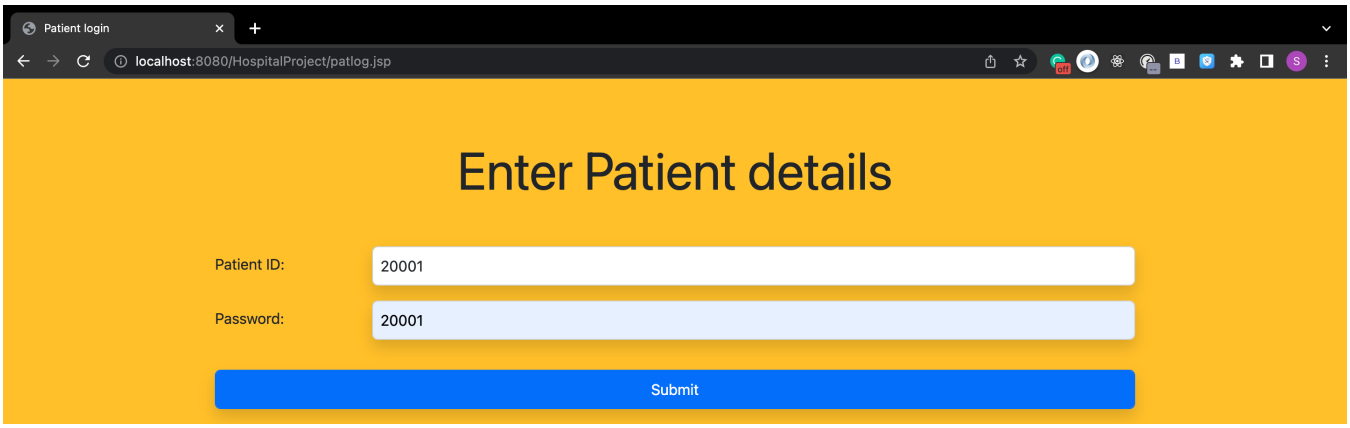

Figure 37: Patient login page(using correct credentials)

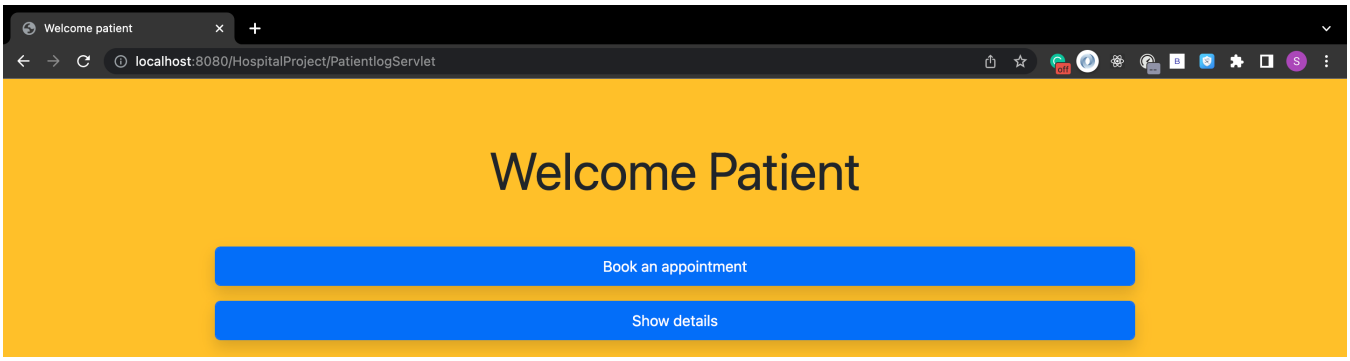

Figure 38: Page after successful patient login

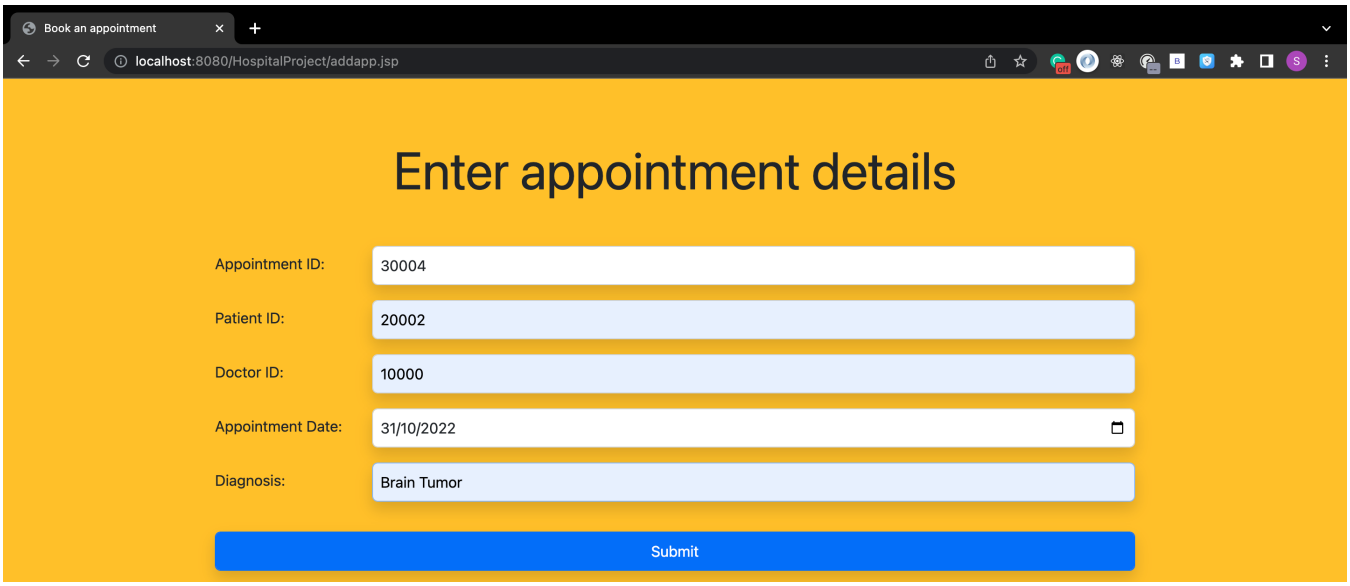

Figure 39: Booking an appointment (by patient)

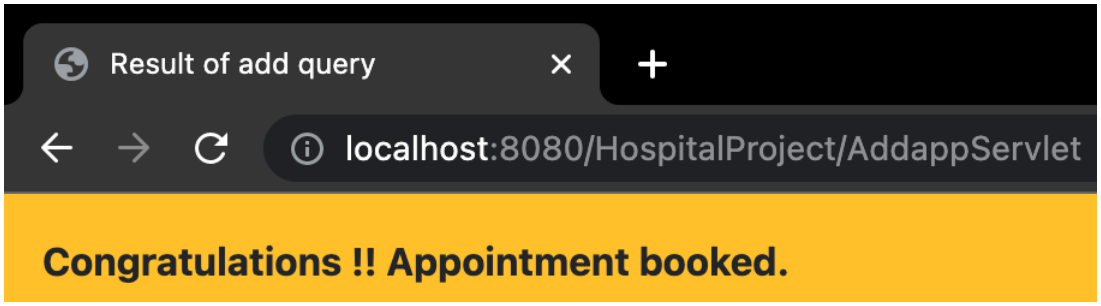

Figure 40: Successful booking

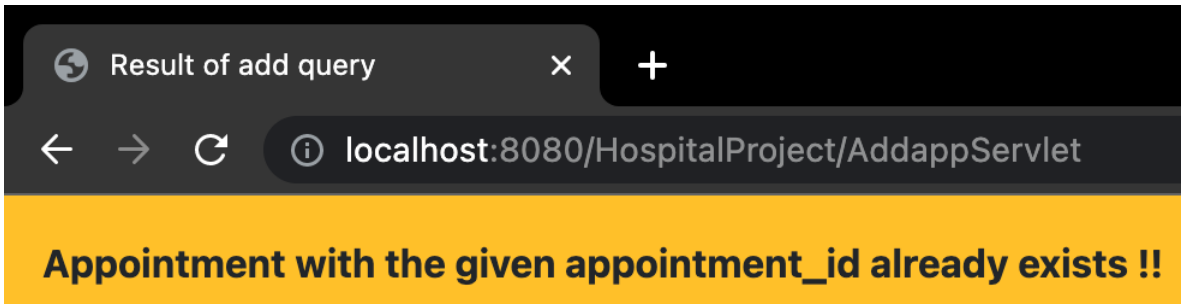

Figure 41: Error message if we try to violate the PK constraint

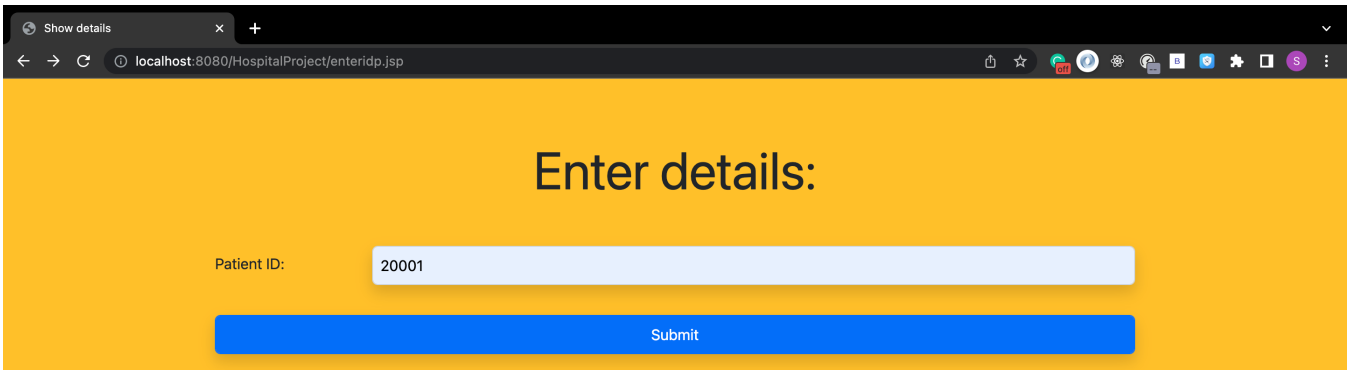

Figure 42: Page for entering id for showing details

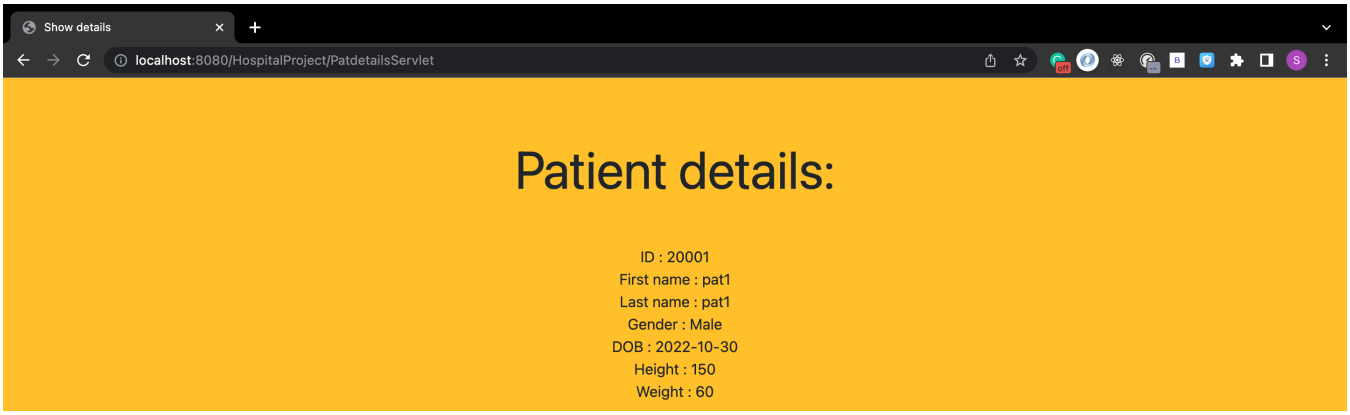

Figure 43: Viewing patient details

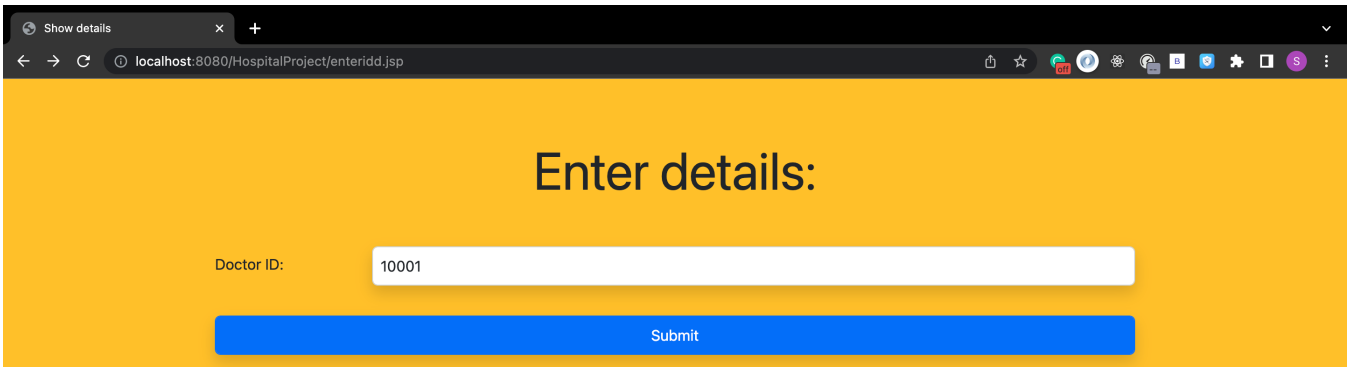

Figure 44: Page for entering id for showing details

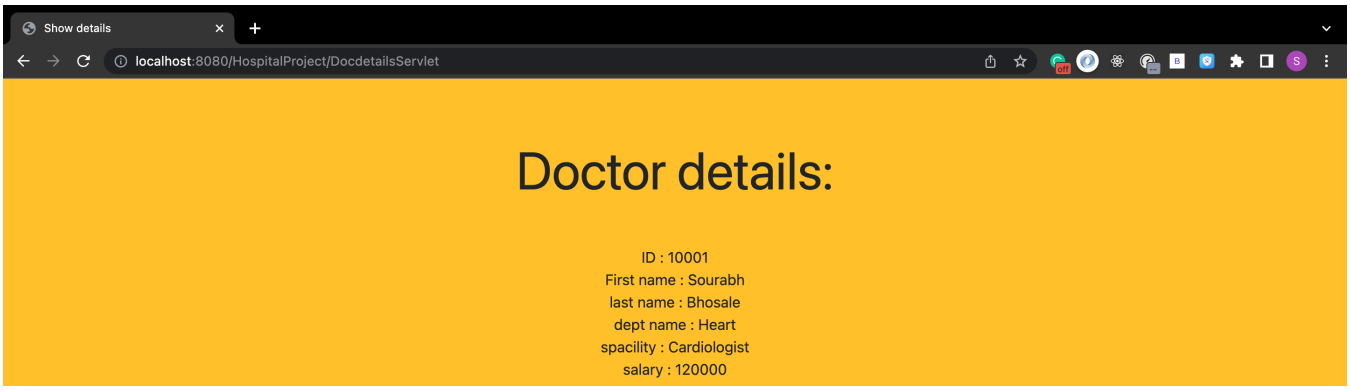

Figure 45: Viewing doctor details

8 Conclusion

We created an hospital management system that stores hospital data about patients, doctors, staff, appointments , departments and the credentials of involved parties in MySQL database and used JDBC to communicate with MySQL database from Servlets to make our website interactive. This can be used to manage a hospital or a clinic kind of setup that will provide patient with the flexibility to book appointment both with admin (the receptionist at the hospital can act as admin) and at the ease of their home.