Hospital Management System

The COVID-19 virus is spreading havoc around the world, and hospitals around the world need to plan for a separate division to handle the growing number of cases. So, my project helps to manage the activities of the COVID-19 department, along with helping patients access their data and test results easily and securely. Doctors can use this system, to access patient records along with other useful data. They can book a room easily using the “book room” functionality.

ReadME

Welcome to my project on hospital management system for covid-19 patients.
In this file, I have defined clear steps to run the application.

Steps

1) Create a fresh folder and name it anything you like.

2) Unzip the application code file using any software of your choice. On unzipping make sure you have only the code files in the directory. (Unzip choudhari_project.zip)

3) Install node js on your local system from https://nodejs.org/en/download/

4) To check if node has been successfully installed, run the command "node -v" in the terminal to see the version node installed. If there is no output then there was an issue while installing node. Check on google for the error or contact me if there is an issue.

========================= OPTIONAL NOT REQUIRED ==============
Extra steps optional or if some error in packages is encountered

a) Install npm by typing the command "npm install -g npm" in the terminal.

b) Check if it has successfully installed by running the command "npm -v" in the terminal. Again if there is an error, check on google for the error or contact me if there is an issue.

c) That is all we need to download :) Now, type the command *npm install* in the terminal and wait for all the packages to be downloaded and installed. Done, the setup has finished.
==

5) *Warning*: Before running the code, make sure the appropriate database (hospital) has been created in MYSQL 8.0.22. There will be an issue if MYSQL 5 or any other is used. Just execute the hospital_final_reports.sql file in the MYSQL Workbench, and make sure that the hospital database has been created.

IMportant Step -> If not done code will not work
1) In the dbcon.js file change the username and password of the createConnection function to as per your database credentials.
2) In MYSQL WorkBench or through the shell execute the command
In place of username and password give your own username and password for the database.

```
ALTER USER 'usename'@'localhost' IDENTIFIED WITH mysql_native_password
BY 'password'
```

6) Execute the command "npm start", and wait for the site to be up. In the terminal, you will see the http address on which the site has started. Open the link in a web browser. In the terminal check for the message MYSQL Connected !!!, which indicates that the server has connected with the database. Follow the interaction guide below to get the best out of the system.

Remember: Execute this command where the files are located. (Path where bin,controllers,... folders are located).

Interactions within the site.

In the beginning you are greeted with the homepage.
The user can login with three different roles.

Role 1: Patient
Go to the login page > Enter in the patient creadentials.
As an example I have listed the usernames below and all have a common password for simplicity.
==
Patient Usernames
sampat
adampat
sarahpat
montypat
amypat

Password
qwerty55

==

Once you are logged in, have a look around. There you can access patients "Appointments and Prescription History", "Lab Reports" and "Medical Bills" information. Logout, when done.

Role 2: Doctor
Go to the login page > Enter in the doctor credentials.
As an example I have listed the usernames below and all have a common password for simplicity.

==

Doctor Usernames
lauradoc
jamesdoc
sandydoc
seandoc

Password
qwerty123

==

Once you are logged in, have a look around. There you can access doctors "Patient Records", "Salary Records" and "Book a room".
Disclaimer: For the purpose of demonstration, I have fixed the room booking for the 25th of November, and only for that day. Due to time constraint and other subject project, I was not able to complete it, but in the future work, I will include dynamic room booking functionality that will handle bookings for different days.
Logout, when done.

Role 3: Admin
Go to the login page > Enter in the doctor credentials.
As an example I have listed the usernames below and all have a common password for simplicity.

==

Doctor Usernames
admin

Password
admin

==

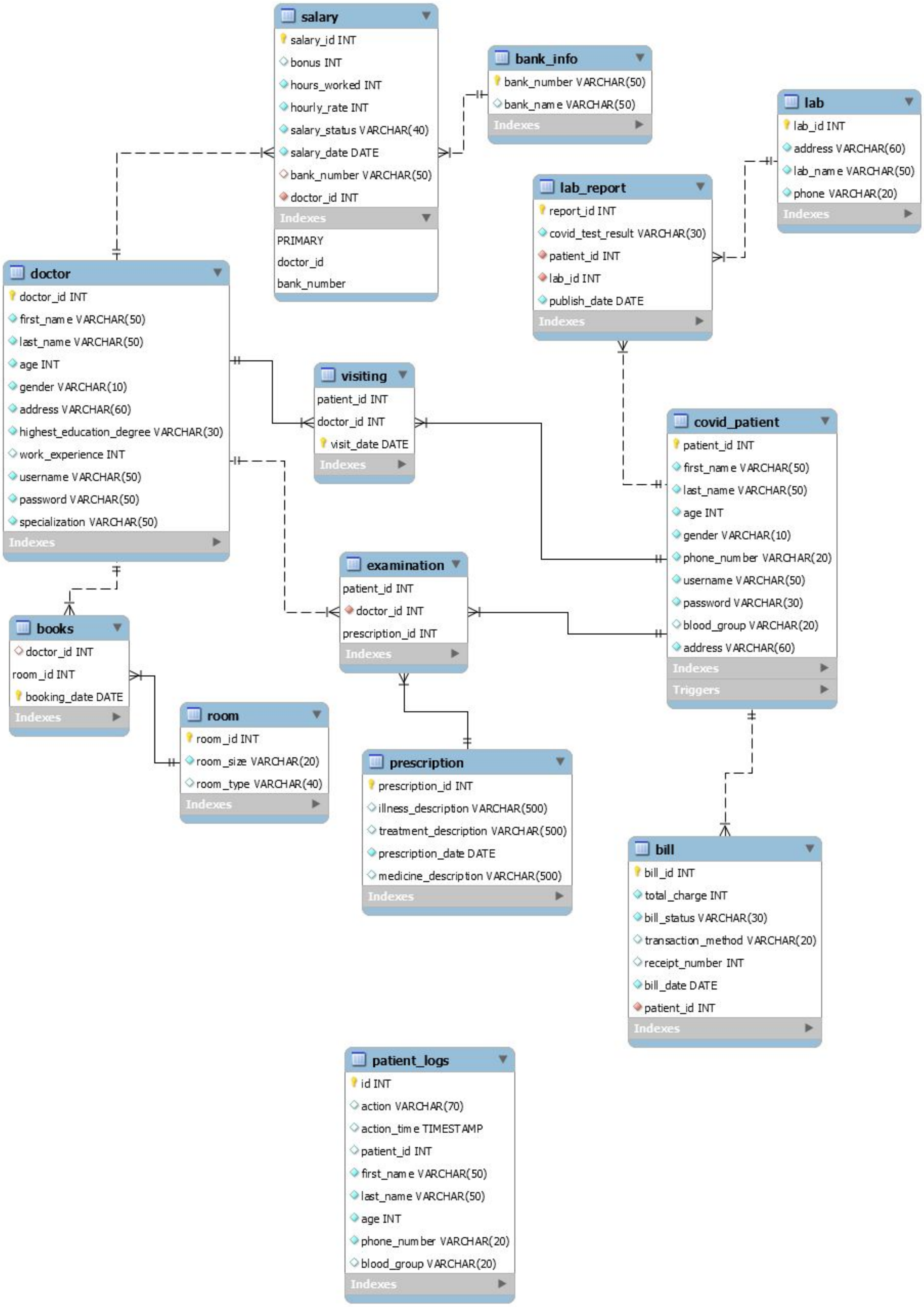
This feature was added just a day before submission, and has the least security functionality. I will keep updating it to make it better. But, on logging in with the admin credentials there is a dashboard that the admin can interact with. The left menu gives the limited but fulfilling options that the admin can perform on the database. Return to the main homepage after finishing the task.

On an endnote, I would like to keep updating this system, to add in many more functionalities, and to improve the system as a whole. Thank you for your time today. Stay healthy, stay safe. Cheers!

Technical Specification

OS: Windows 10
Editor Environment: VS Code
Front End: Bootstrap, HTML, CSS, Javascript.
BackEnd: Express, NodeJS.
Database: MYSQL
Browser: Chrome

UML Diagram


salary
salary_id INT
bonus INT
hours_worked INT
hourly_rate INT
salary_status VARCHAR(40)
salary_date DATE
bank_number VARCHAR(50)
doctor_id INT
Indexes
PRIMARY
doctor_id
bank_number
bank_info
bank_number VARCHAR(50)
bank_name VARCHAR(50)
Indexes
lab
lab_id INT
address VARCHAR(60)
lab_name VARCHAR(50)
phone VARCHAR(20)
Indexes
lab_report
report_id INT
covid_test_result VARCHAR(30)
patient_id INT
lab_id INT
publish_date DATE
Indexes
doctor
doctor_id INT
first_name VARCHAR(50)
last_name VARCHAR(50)
age INT
gender VARCHAR(10)
address VARCHAR(60)
highest_education_degree VARCHAR(30)
work_experience INT
username VARCHAR(50)
password VARCHAR(50)
specialization VARCHAR(50)
Indexes
visiting
patient_id INT
doctor_id INT
visit_date DATE
Indexes
covid_patient
patient_id INT
first_name VARCHAR(50)
last_name VARCHAR(50)
age INT
gender VARCHAR(10)
phone_number VARCHAR(20)
username VARCHAR(50)
password VARCHAR(30)
blood_group VARCHAR(20)
address VARCHAR(60)
Indexes
Triggers
examination
patient_id INT
doctor_id INT
prescription_id INT
Indexes
books
doctor_id INT
room_id INT
booking_date DATE
Indexes
room
room_id INT
room_size VARCHAR(20)
room_type VARCHAR(40)
Indexes
prescription
prescription_id INT
illness_description VARCHAR(500)
treatment_description VARCHAR(500)
prescription_date DATE
medicine_description VARCHAR(500)
Indexes
bill
bill_id INT
total_charge INT
bill_status VARCHAR(30)
transaction_method VARCHAR(20)
receipt_number INT
bill_date DATE
patient_id INT
Indexes
patient_logs
id INT
action VARCHAR(70)
action_time TIMESTAMP
patient_id INT
first_name VARCHAR(50)
last_name VARCHAR(50)
age INT
phone_number VARCHAR(20)
blood_group VARCHAR(20)
Indexes

User Flow

The user first visits the website. On landing on the home page, they can choose to login. Based on the type of the user, different users can log in to their accounts. A patient can log in to access his personal information. Similarly, doctors can access their own and patient data. There is one more extra user, who is the super user, who has a simple and easy interface to add/remove items in and out of the database. On completion, they choose to logout, and return back to the homepage.

Lessons Learned

1) Technical Expertise Gained

The project has been a challenging task to do and I have gained a lot of technical expertise. Firstly, I got to understand the core concepts of the database, and designing a database is the critical step to the entire project. Database design had many challenging hurdles, but the knowledge I gained in this course helped me overcome them. Making a website was another difficult task. I have not built a website before in its entirety, just bits and pieces, therefore learning NodeJS and other essential technologies gave me a hands on experience to implement my learned concepts, and to build a website as a whole.

2) Insights

The project has helped me gain a clear understanding of, how a hospital management system functions? In a large hospital, there are more complex databases and systems, but this project has taught me the skills and vision to work on such complex designs. Time management was another useful life lesson I gained. Working daily for 4hrs on the project has instilled the notion of optimizely performing tasks and saving time for other complicated tasks. Therefore, I was able to bring out the most in the short window of 4hrs.

3) Alternate Design Ideas

At first, I weighed on using MongoDB as the database, but found myself not gaining any use in getting a hands on experience on relational databases. Therefore, to experiment with the concepts that I learned in the database course, I decided to scrap that idea. Alternative, front end decision was to make it in React, but I found it difficult to learn a new technology in such a short span, therefore I went ahead with using base HTML, CSS, JS.

4) Code Not Working

I have written error handling methods for all the functionalities, so the code might not break. I have not got the time to implement a secured login for the admin user, as I decided to add the user at the end moment. But, in all all the sections are functioning as intended. I have also fixed the date to 25th November, but will work further on adding a dynamic room booking feature for all the dates. The footer is not at the bottom, when seen on a large screen. There is white space after it.

Future Work

1) Planned Use of the database

I plan to keep improving the database and to host the site on a cloud provider, like AWS. I will also collaborate with more qualified subject experts to improve the database design.

2) Potential Areas of added functionality

The scope of improvement is large, and there are many more functionalities that can be added and improved. I have listed some ideas below.

a) Adding a payment mechanism for the billing section.
b) Making the website more aesthetically pleasing.
c) Adding a time components to the room booking feature
d) Clean the code, and write more detailed test cases.
e) Adding more functionalities for the patient, like giving access to book an appointment.
f) Adding complex security, right now it is at the minimum level.

There are many more, and will add them as time progresses.

3) Future Use

I plan to host the website on a cloud service so that everybody can access the website. Further improvements as listed above, might make it a fully functional website that clients can use for the hospital management.

Conclusion

The knowledge I have gained through the course and the project will help me apply for more skilled opportunities in the future. Understanding the intrinsic details of the database design process has opened a whole new level of possibilities and exploration to do. I have learned the process of designing a database, gained domain knowledge in the field of hospital management, and learned to build a website. I would like to thank the Professor for being there to guide me in every single step of the project.